Marcelo André Ackermann

Aderência de Controles de Acesso em SGBDs Relacionais às Políticas de Segurança de Aplicações.

Florianópolis, Outubro de 2003.

UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Marcelo André Ackermann

Aderência de Controles de Acesso em SGBDs Relacionais às Políticas de Segurança de Aplicações.

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação.

Murilo S. de Camargo

Florianópolis, Outubro de 2003.

Aderência de Controles de Acesso em SGBDs Relacionais às Políticas de Segurança de Aplicações.

Marcelo André Ackermann

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação Área de Concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

_____
Prof. Dr. Raul Sidnei Wazlawick

Banca Examinadora

_____
Prof. Dr. Murilo S. de Camargo (orientador)

_____
Prof. Dr. Rosvelter J. Coelho da Costa

_____
Prof. Dr. Mario A. R. Dantas

_____
Prof. Dr. Vitório B. Mazzola

Este trabalho é dedicado a minha esposa Andréia Aline e ao meu filho Álvaro Afonso, pela compreensão e apoio durante toda jornada do mestrado, pois foram incentivos na conclusão do mesmo.

Resumo

A fraude de informação tornou-se um problema em escala mundial, causando muito prejuízos para empresas, e na maioria das vezes, este problema surge dentro das organizações. Com o aumento de dados, e o crescimento do número de usuário que dependem da informação nas empresas, controlar o acesso, se tornou ponto chave para segurança e proteção, de segredos de negócios, estratégias comerciais ou até mesmo na proteção do capital intelectual, garantindo o bom funcionamento dos aplicativos e SGBD.

Dentro deste aspecto, o objetivo desta dissertação está centrada na análise técnica, da aderência do controle de acesso aos dados dos SGBD's relacionais, em relação às políticas de seguranças exigidas pelas aplicações. Dentro deste contexto, levantou-se os fundamentos de segurança das informações e políticas de controle, para servir de base e sustentação da pesquisa e levantamento dos controles de acesso disponíveis nos SGBD's, apresentando seus benefícios e suas vulnerabilidades.

Na busca da possível validação da hipótese de centralização dos controles de acesso no SGBD e a aderência do mesmo às políticas de segurança, pesquisou-se em detalhe os mecanismos de controle de acesso do SGBD Oracle por este ser considerado um dos mais seguros do mercado. Também foram estudadas as pesquisas que estão sendo feitas na área de controle de acesso de dados em SGBD relacional na tentativa de encontrar uma solução para os problemas. Finalizando a dissertação, são apresentadas as análises e os resultados da pesquisa na busca de exemplificar as hipóteses levantadas.

Abstract

The information fraud became a problem in world-wide scale, causing many damages for companies, and most of the times, this problem appears inside of the organizations. With the increase of data, and the user's growth number that depend on the information within the companies, to control the access, has become a key factor for security and protection, of business-oriented secrets, commercial strategies or even though in the protection of the intellectual capital, guaranteeing the good functioning of applications and the SGBD.

Inside of this aspect, the objective of this paper, is centered in the technical analysis, of the control of access to the relationary data of the SGBDs, in relation to the policies of securities demanded by the applications. In this context, fundamentals of security had been researched of the information and policies of control, cientifically to serve as base and sustentation of the research and survey of the available controls of access in the SGBDs, presenting its benefits and vulnerabilities.

In the search of the possible validation of the hypothesis of centralization of the controls of access in the SGBD and the use of the same one to the security politics, it was searched in detail the mechanisms of control of access of the Oracle SGBD, considered one of the safest. Also the research had been studied that is being made in the area of control of access of data in relationary SGBD in the attempt to find a solution for the problems. Finishing the paper, the analyses and the results of the research in the search of producing samples the raised hypotheses are presented.

SUMÁRIO

LISTA DE FUGURAS

LISTA DE QUADROS

LISTA DE ANACRÔNICOS

BD – Banco de dados;

DAC – Discretionary access control – Controle de acesso discriminatório;

DBA - Administrador da base de dados;

DDL - Data dictionary language – Linguagem de definição de dados;

DML – Data manipulation language – Linguagem de manipulação de dados;

eBusinesses – Sistema de negocio via Web;

ERP - Enterprise Resource Planning – Planejamento dos recursos de uma empresa;

GRANT – Comando usado na linguagem SQL para conceder privilégios;

LBAC - Lattice-based access control – Controle de acesso baseado em grade ou rótulo;

LDAP- based - Lightweight Directory Access Protocol;

MLS – Multi Level segurity – Segurança em multi nível;

NPD - Named protection domain – Dominio de proteção nomeada;

RBAC – Role-Based Access Control – Controle de acesso baseado em papéis;

REVOKE – Comando usado na linguagem SQL para revogar privilégios;

SGBD – Sistema gerenciador de base de dados;

SQL – Structured Query Language – Linguagem de construção de consulta;

VIEW – Visões da base de dados;

VPD - Virtual Private Database – Banco de dados virtual privado;

Web - www ou World Wide Web – Rede mundial de computares.

1 INTRODUÇÃO

Os avanços tecnológicos têm proporcionado às empresas maior eficiência e rapidez na troca de informações e tomada de decisões. Computadores, cada vez mais rápidos, são lançados em curto espaço de tempo. A internet, agora disponível para todos, tem permitido às empresas praticar o Comércio Eletrônico.

Com o uso cada vez maior da internet, e de ferramentas estratégicas os sistemas gerenciadores de bancos de dados assumem um papel fundamental nos sistemas das empresas. Por outro lado, agentes ameaçadores aos ambientes computacionais estão em constante evolução. Com podemos perceber, todas estas tecnologias e avanços têm colocado muitas empresas em uma posição delicada em alguns casos. Problemas de origem interna e externa têm marcado presença no dia-a-dia, principalmente nas empresas que não possuem políticas de segurança definidas e implementadas.

Dentro deste contexto, o sistema gerenciador de banco de dados assume cada vez mais um papel importante, onde o mesmo deve permitir que usuários compartilhem seletivamente dados, tendo a habilidade para restringir o acesso, devendo prover e manter mecanismos de proteção e segurança (GRIFFITHS,1976).

Segurança de dados é uma função importante de um sistema gerenciador de banco de dados, que é de, proteger os dados contra acesso sem autorização, destruição maliciosa ou introdução e alteração acidental de inconsistência. Perdas acidentais de consistência geralmente são causadas por anomalias e erros no sistema, estas são mais fáceis de serem resolvidas. Por outro lado, acessos maliciosos já são mais difíceis de

serem tratados, pois estes podem ser resultados de leituras sem autorização, geralmente causando modificação e destruição de dados (PISSINOU,1994).

Como hoje as informações se tornaram os bens mais valiosos para empresa, surgem a necessidade, cada vez maior, de controles de acesso a estes dados. Os mecanismos de controle de acesso de preferência devem ser centralizados para facilitar sua administração, visto que o volume de informação vem crescendo dia-a-dia, o que dificulta o processo de garantir a segurança.

Estão sendo desenvolvidos módulos de controle de segurança nas aplicações na tentativa de eliminar este problema, mas muitas vezes trazendo outros. Pois, as políticas de acesso se alteram com o passar do tempo, obrigando manutenções nestas aplicações, que geralmente tem o controle em código fixo na aplicação.

Surgindo o questionamento: Os controles de acesso dos SGBD's são flexíveis e aderentes totalmente às políticas de seguranças exigidas pelas aplicações?

- Como são controles de acesso SGBD's?
- Estão de acordo com as exigências das aplicações?
- Possuem flexibilidade e adaptam-se as mudanças exigidas pelas empresas e mercado?
- Como é feita concessão de privilégio?
- A concessão é fácil em grandes volumes de dados?
- Qual a granulação dos controles de acesso?

1.1 MOTIVAÇÃO

O crescente aumento do volume de informações disponíveis para as empresas, está garantindo o sucesso ou o fracasso das mesmas. Com isso, aumentando cada vez mais a responsabilidade dos administradores de dados em manter mecanismos e meios de controle de acesso afinados com as políticas de segurança das aplicações das empresas.

Quanto maior o volume de dados, mais complexos torna-se os controle de acessos aos mesmos, visto que, conseqüentemente o número de usuário e as formas de acessar se multiplicam com o passar dos tempos, principalmente com a interoperabilidade dos aplicativos com os sistemas gerenciadores de base de dados.

Surgindo à necessidade de melhores controles de acesso, os quais flexibilizem e facilitem o trabalho de concessão de privilégios aos usuários dos sistemas. Com isso, percebe-se à necessidade de avaliação e adequação dos dois aspectos: os controles de acessos dos sistemas gerenciadores de base de dados e as políticas de segurança das aplicações, numa perspectiva comparativa, de modo a analisar de forma técnica a possibilidade da aderência dos controles de acessos dos SGBD's as políticas de segurança das aplicações e quem sabe a centralização destes controles.

1.2 OBJETIVOS

O objetivo do trabalho é analisar a aderência dos controles de acessos dos sistemas de banco de dados com as políticas de segurança nas aplicações. Para isso, devemos pesquisar e levantar como são os controles de acesso dos SGBD's. Verificando se estes controles estão de acordo com às exigências das aplicações, comprovando a existência de flexibilidade e adaptabilidade do mesmo às exigências do mercado. Dentro do processo de liberação de concessão de privilégios, levantar as formas e as facilidades dos mesmos. Verificando e analisando se a granulação dos controles está de acordo com as políticas de seguranças das aplicações.

O presente trabalho enquadra-se em num projeto mais ambicioso de desenvolvimento de mecanismos de controle de acesso, que possibilitem a centralização total destes controles em um único local, tirando a preocupação dos desenvolvedores em segurança, deixando esta para os responsáveis por segurança das empresas.

1.3 ESTRUTURA DA DISSERTAÇÃO

A dissertação compreende mais cinco capítulos para além deste primeiro de caráter introdutório.

No segundo capítulo, são descritos os conceitos básicos de segurança, aspectos relevantes da segurança de dados, em seguida descreve os tipos de controles, política de segurança e para concluir determina o problema base.

O terceiro capítulo, contempla uma descrição dos controles de acesso padrões do SGBD, apresentando os controles semânticos, o que são privilégios, e quais os tipos disponíveis de concessão e como funciona o processo de concessão de privilégios. E, ainda, apresenta uma descrição dos modelos de controle de acessos disponíveis e suas granulação de segurança, como se procede as revogações, e os papéis que pode ser controlados pelos SGBD's.

O quarto capítulo, é dedicado a apresentar os controles de acesso de um SGBD específico, para o qual foi escolhido o Oracle por se considerado pelo mercado um dos mais completos em termos de controles de acesso aos dados, onde neste foram estudados quais os tipos de privilégios disponíveis, como estes podem ser usados e mantidos.

Descreve também a forma de usar papéis para administrar privilégios, apresenta quais os tipos de papéis disponíveis, comenta como pode ser usado stored procedures no SGBD para administrar privilégios. Aspectos de segurança ao nível de registros, como pode ser conseguido uma granulação fina no controle de acesso, utilizando as extensões desenvolvidas sobre SQL.

No quinto capítulo, são apresentadas pesquisas realizadas nas áreas de controle de acesso de segurança de dados. São descritas pesquisas feitas em controle discriminatório e controle obrigatório.

No sexto capítulo, analisa-se a aderência dos tipos de controles de segurança dos SGBD aos tipos de políticas de segurança das aplicações, embasados nos referenciais teóricos levantados. Discute as perguntas levantadas na definição do problema.

No último capítulo, são retiradas as conclusões do presente trabalho e referidas algumas perspectivas de desenvolvimento futuro.

2 CONCEITOS BÁSICOS DE SEGURANÇA

2.1 INTRODUÇÃO

Este capítulo tem como intuito mostrar os conceitos relevantes da segurança de informação no que diz respeito às aplicações comerciais, bem como, descrever a complexidade que é o processo de concessão privilégios e o desenvolvimento de uma aplicação quanto à segurança.

2.2 SEGURANÇA DA INFORMAÇÃO

Segurança é um termo tão genérico que é melhor pensarmos em aspectos de segurança da informação. Em vez de perguntar quanto queremos de segurança, devemos perguntar quanto queremos de disponibilidade do sistema, ou qual a necessidade de confidencialidade, por exemplo. Existem vários aspectos de segurança, sendo que três são considerados centrais ou principais (ALBUQUERQUE,2002).

- Confidencialidade.
- Integridade.
- Disponibilidade.

2.2.1 Confidencialidade

Capacidade de um sistema de impedir que usuários não autorizados vejam determinada informação, ao mesmo tempo em que usuários autorizados podem acessá-la (ALBUQUERQUE,2002).

A confidencialidade é a propriedade que visa manter o sigilo, o segredo ou a privacidade das informações evitando que pessoas, entidades ou programas não autorizados tenha acesso às mesmas (MOREIRA,2001).

Dentro do aspecto de confidencialidade é difícil controlar o sigilo quando dois usuários do sistema têm acesso à mesma informação em uma tabela do sistema gerenciador de base de dados, mesmo que cada um destes tenha direito de acessar parte da informação.

Um sistema seguro assegura a confidência de dados. Isto significa que indivíduos apenas podem ver os dados que é permitido ele ver. Confidencialidade tem vários aspectos diferentes, como podemos ver nesta seção.

- Privacidade de comunicações.
- Armazenamento seguro de dados sensíveis.
- Autenticação de Usuários.
- Controle de acesso granular.

2.2.1.1 Privacidade de Comunicações

Como você pode assegurar a privacidade de comunicações de dados? Privacidade é um conceito muito amplo. No mundo empresarial, privacidade pode envolver segredos de comércio, informação proprietário sobre produtos e processos, análises competitivas, como também, planos de vendas e comercialização. Para governos, envolve privacidade, tal como, a habilidade colecionar e analisar informação

demográfica, mantendo e protegendo a confidencialidade de milhões de cidadãos. Também envolve a habilidade de manter segredos que afetam os interesses do país.

2.2.1.2 Armazenamento seguro de dados sensíveis

Como você pode assegurar aqueles dados permanecem privados, uma vez foi armazenado? Dados confidenciais quando armazenados, devem ser protegidos sua integridade e privacidade nos bancos de dados e servidores onde colocados.

2.2.1.3 Autenticação de Usuários

Como você pode determinar quais pessoas das organizações têm o direito para ver quais dados? A autenticação é um modo de garantir que o usuário que acessou é realmente quem ele diz ser. Métodos de autenticação buscam garantir a identidade de usuários nos sistemas: confirma que uma pessoa é quem diz ser, e não um impostor. Por não fazer parte do escopo deste trabalho não se detalhará este item.

2.2.1.4 Controle de acesso granular

Quantos dados um usuário particular deveria ver? Controle de acesso é a habilidade isolar porções do banco de dados, de tal forma que o acesso aos dados não se torne aquele acesso onde todos tem acesso ou nenhum tem. Um secretário no departamento de Recursos Humanos poderia necessitar ter acesso aos dados de empregados que estão na tabela “emp”, mas, o mesmo não deveria ter acesso às informações de salário da companhia inteira. A granulação do controle de acesso é o grau que pode ser diferenciado no acesso aos dados em tabelas particulares, visões, registros, e colunas de um banco de dados.

Observe que existe uma distinção entre autenticação, autorização, e controle de acesso. Autenticação é o processo pelo qual a identidade de um usuário é conferida.

Quando um usuário é autenticado, ele é verificado como um usuário autorizado de uma aplicação. Autorização é o processo pelo qual os privilégios do usuário são averiguados. Controle de acesso é o processo pelo qual o acesso do usuário para dados físicos na aplicação está limitado, baseado nos privilégios dele. Estes são assuntos críticos em sistemas distribuídos. Por exemplo, se João estiver tentando ter acesso no banco de dados, a autenticação o identificaria como um usuário válido. A autorização verificaria o direito dele de conectar-se ao banco de dados com privilégios de Gerente de Produto. O controle de acesso na sessão de usuário obrigaria o mesmo ter o privilégio de Gerente de Produto.

2.2.2 Integridade

Consiste em proteger a informação contra qualquer tipo de alteração sem a autorização explícita do autor da mesma (MOREIRA,2001).

A integridade é um atributo de uma informação que indica que esta não foi alterada ou, se foi, o foi de forma autorizada; capacidade de um sistema de impedir que uma informação seja alterada sem autorização ou, ao menos, de detectar se isso ocorreu (ALBUQUERQUE,2002).

A perda de integridade pode ser intencional ou não. Independente da forma ou motivo, a questão é: Quanto a empresa vai gastar para recuperar ou reconstruir os dados? Sem dúvida nenhuma, o problema da perda da integridade das informações pode ser catastrófico para qualquer empresa.

2.2.3 Disponibilidade

Indica a quantidade de vezes que o sistema cumpriu uma tarefa solicitada sem falhas internas sobre o número de vezes em que foi solicitado a fazer uma tarefa (ALBUQUERQUE,2002).

De acordo com (MOREIRA,2001), os esforços da empresa em proporcionar a disponibilidade dos seus recursos, sejam eles sistemas, informações ou processos, ocorrem quando estes necessitam de acesso contínuo e ininterrupto.

Sobre este aspecto deve-se procurar soluções que atendam esta necessidade, disponibilizando a informação para a pessoa certa e no momento em que ela precisar.

2.3 SEGURANÇA NAS APLICAÇÕES COMERCIAIS

O desenvolvimento de sistemas é um dos pontos mais críticos para a garantia de segurança, da informação em uma empresa. Procedimentos, treinamento, política de segurança, dentre outros, são aspectos importantes a serem considerados (ALBUQUERQUE,2002).

Em uma aplicação comercial, seja ela centralizada, cliente/servidor ou múltiplas camadas, existem vários recursos que devem ser protegidos para que somente usuários autorizados possam acessá-los. Entre eles podemos citar a própria aplicação, conexões com bancos de dados (BD's), componentes da aplicação, regras e transações de negócio, informações, diretórios e arquivos da rede. A implementação do acesso a esses recursos é feita em conjunto por três gerenciadores de segurança: O sistema operacional de rede, o sistema gerenciador de DB e a aplicação (SQUADRA,2002).

Um processo de desenvolvimento bem fundamentado e estruturado, com atenção específica aos aspectos de segurança, aliado ao uso de ferramentas adequadas e seguras, é a melhor arma que se tem contra os prejuízos decorrentes de falhas de segurança em software (ALBUQUERQUE,2002).

Nenhum processo garantirá a eliminação de todos os erros do desenvolvimento de sistemas, nem a produção de um software totalmente seguro. Mas, o que vemos hoje, principalmente em sistemas desenvolvidos especificamente para uma empresa ou para um número reduzido de usuários, são falhas grotescas de segurança (ALBUQUERQUE,2002).

2.4 CONTROLE DE ACESSO À APLICAÇÃO

O controle de acesso do sistema já começa na própria aplicação, em seus arquivos e diretórios. Pode-se fazer a segurança através do sistema operacional de rede, não permitindo que alguns usuários tenham acesso ao arquivo executável e demais arquivos da aplicação. A dificuldade dessa implementação é que o acesso da aplicação fica a cargo do administrador de rede, podendo ficar sobrecarregado caso exista uma constante mudança de pessoas na empresa. Caso esse recurso seja implementado através de uma aplicação, os próprios gerentes da área poderiam proibir ou permitir seu pessoal de acessar uma ou mais aplicações.

Além disso, para implantar esse tipo de segurança através da rede, seria necessário que os arquivos executáveis ficassem na rede. Para obter uma melhor performance do sistema e não trafegar o executável pela rede, algumas empresas optam por colocar seus executáveis em cada máquina-cliente e tentam utilizar softwares de distribuição para facilitar a implantação. Nesse tipo de configuração, seria difícil fazer o controle via sistema operacional, sendo extremamente útil um controle via aplicação (SQUADRA,2002).

2.5 CONTROLE DE ACESSO A CONEXÕES

A conexão ao Sistema gerenciador de base de dados (SGBD) é um outro recurso interessante.

Isso acontece quando uma mesma aplicação pode ser utilizada para acessar diferentes bases de dados. Como exemplo, podemos citar uma aplicação que poderá ser usada para acessar um banco de desenvolvimento, um banco de homologação e um banco de produção, permitindo que o usuário possa escolher em qual ele irá conectar no momento que entrar na aplicação. Outros exemplos são aplicações que possam acessar, dependendo do usuário, um BD da matriz ou um BD de uma filial. Pode-se proibir e permitir o acesso dos usuários ao BD através do SGBD. Entretanto, isso não é suficiente

para automatizar o processo de configuração sem ter a necessidade de construir uma aplicação para cada base de dados. É preciso que a própria aplicação identifique as conexões que o usuário possa optar (SQUADRA,2002).

No SGBD as tabelas da aplicação devem ter um dono (*owner*), então para que o usuário possa acessar as tabelas de um determinado dono, este deve ter recebido tal direito, este controle é feito ao nível de SGBD, este controle será apresentado no próximo capítulo. Outra forma de fazer a conexão, é quando a aplicação utilizar, sempre o usuário dono, fazendo ela própria o controle de usuário e senhas, desta forma garante que o usuário não consegue acessar as informações a não ser utilizando a aplicação, podendo esta controlar o acesso as informações na horizontal e na vertical.

Mas, quando o usuário necessita de informações mais gerencias que o sistema não fornece este deve acessar as tabelas diretamente com outros aplicativos, voltando assim o problema de falta de controle ao nível de colunas e linhas nas tabelas. Isso apenas é conseguido utilizando visões (*views)* do BD.

2.6 CONTROLE DE ACESSO A REGRAS E TRANSAÇÕES DE NEGÓCIO

Chamamos de regras e transações de negócio, a lógica da aplicação que vai desde a mais básica como incluir, alterar e excluir dados em uma tabela, até transações mais complexas. Grande parte dessas regras é operações com a base de dados e devem também poder sofrer restrições de acesso e execução.

Algumas empresas optam por implantar essas regras no próprio BD através de eventos (*triggers)* e procedimentos armazenados (*stored procedures).* Desta forma, é possível utilizar os recursos do SGBD para prover o controle de acesso. Nesse tipo de implementação, o usuário receberia mensagens do BD restringindo seu acesso a determinados objetos. Entretanto, essas mensagens teríam que ser tratadas dentro da aplicação, de tal forma, que o usuário consiga amigável identificar o erro e conseqüentemente não ter acesso ao determinado recurso. Muitas vezes, é mais

interessante desabilitar um recurso visual na aplicação para que o usuário não tente executar uma opção, a qual ele não esteja autorizado.

Outras empresas optam por implantar todas as regras ou parte delas na própria aplicação. Neste tipo de cenário, um controle de acesso pela aplicação se torna indispensável, por ser insuficiente proibir o usuário de incluir, alterar ou excluir registros de uma tabela. Se o usuário, por exemplo, tem autorização de inserir em uma tabela, ele pode também entrar em qualquer ferramenta que acesse o BD e incluir um registro na tabela. Mas, o que acontece, caso exista uma regra de negócio que não permita que o registro seja inserido na tabela, sem que as demais operações sejam feitas em uma mesma transação, mantendo a integridade lógica dos dados da aplicação?'

Como exemplo, pode-se citar a baixa de um estoque sendo feita simultaneamente com a inclusão do pedido. O usuário não pode simplesmente incluir o pedido sem ao mesmo tempo baixar o estoque. O acesso às tabelas não garante que ele irá executar as duas operações simultaneamente. Portanto, neste caso, é comum que as senhas dos usuários possam apenas ler a base de dados e todas as alterações ficam para uma senha de maior acesso que será utilizada pela aplicação e que irá controlar o acesso às regras de negócio, ou que o usuário apenas tenha o direito ao acesso quando estiver executando determinada aplicação.

2.7 CONTROLE DE ACESSO A INFORMAÇÕES

Outro tipo de acesso à aplicação que necessita de restrição é o acesso às informações do sistema. Nem todos os usuários podem acessar qualquer informação que esteja na base de dados. Para fazer isso, pode-se restringir o acesso à leitura de determinadas tabelas do BD ou criar visões para cada usuário ou conjunto de usuários, delimitando o que cada um pode ver.

Para o usuário acessar diretamente a base de dados por outra ferramenta, este recurso é viável. Entretanto, a construção de visões pode tornar o processo de

desenvolvimento de sistemas bem improdutivo, já que para cada usuário seriam necessárias visões diferentes para acessar o BD.

Para este tipo de acesso aos sistemas de informação, pode ser mais fácil e produtivo trabalhar nos componentes visuais da aplicação. Assim, pode-se permitir, ou não, que o usuário acesse uma determinada tela de informações, ou até mesmo impedi-lo de ver um determinado campo.

Outro tipo de acesso, às vezes necessário, é a implementação de filtros na consulta, restringindo os registros que o usuário possa ver. Esse filtro pode, por exemplo, ter uma relação direta ou indireta com o usuário que está acessando a aplicação. Nestes casos, o usuário deve estar cadastrado em uma tabela da aplicação para que o filtro seja feito como uma regra de negócio normal que a aplicação terá que seguir.

Mas, estas alternativas só conseguem impedir o acesso indevido às informações caso o usuário apenas consulte às informações via aplicação, o que em alguns casos é impraticável, e dependem de programação prévia.

2.8 CONTROLE DE ACESSO A COMPONENTES

Pode acontecer da empresa querer restringir o acesso a componentes da aplicação, além dos identificados nos itens à cima. Um ou outro usuário não pode ver, editar ou acessar um determinado componente da aplicação. Esse tipo de controle só é possível através de implementações feitas nas próprias aplicações (SQUADRA,2002).

2.9 CONTROLANDO O ACESSO PELA APLICAÇÃO

Como foi visto, grande parte do acesso deve ser controlado pela aplicação. Entretanto, estas implementações são comuns e pode ser reunida de forma a constituir um único modelo de controle de acesso. Não seria interessante que para cada aplicação

um controle de acesso diferente fosse implementado através de soluções isoladas e individuais que não teriam nenhum padrão, dificultando sua utilização pelo usuário final. Portanto, um sistema de controle de acesso é visto como mais um módulo do sistema que deve ser implementado, tendo os seguintes componentes:

Modelo de dados: tabelas que são agregadas ao sistema para armazenar as informações dos recursos e seus acessos.

Aplicação para gerenciar a segurança: aplicação que cadastra e registra os acessos aos usuários dos componentes da aplicação.

Bibliotecas e componentes: funções e componentes que consultam o acesso e fazem às restrições aos recursos da aplicação. Estes componentes podem também auxiliar o processo de cadastramento dos recursos.

2.10 POLÍTICA DE SEGURANÇA

O propósito de uma política de segurança é elaborar os três objetivos de segurança genéricos: de confidencialidade, integridade e disponibilidade, no contexto de um sistema particular. Os objetivos genéricos têm usado o termo “impróprio”. Na definição desta, uma declaração de política de segurança consiste em grande parte na definição do que é o significado de “impróprio" para um sistema particular.

O significado de “impróprio" às vezes é designado através de lei, como para segredo clássico do exército e dos setores do governo. Exigências legais e profissionais são aplicados a registros médicos e outras informações pessoais sensíveis sobre indivíduos. Em geral a política de segurança é largamente determinada dentro de uma organização em lugar de impor por mandato de fora, assim, é principalmente na integridade e área de disponibilidade (SANDHU,1993).

2.11 PREVENÇÃO, DESCOBERTA E TOLERÂNCIA

O objetivo de segurança de dados pode ser checado em dois modos distintos:

Prevenção - Prevenção assegura aquelas brechas de segurança que não podem acontecer. A técnica básica é que o sistema examina toda ação e cheque sua conformidade com a política de segurança antes de permitir isto acontecer. Esta técnica é chamada controle de acesso.

Descoberta - Descoberta assegura aquela atividade no sistema que é registrada suficiente em história como um rastro de auditoria, de forma, que uma brecha de segurança possa ser descoberta depois do fato. Esta técnica é chamada auditoria.

Todo sistema emprega alguma mistura destas duas técnicas. Às vezes a distinção entre estas duas técnicas é obscurecida. Por exemplo, considere um sistema que monitora o rastro de auditoria em tempo real, o qual olha para violações de segurança iminentes para os prevenir. Tal sistema é preventivo por natureza, contudo a tecnologia usada é basicamente de detetive. A distinção não é bastante útil. Nosso foco neste trabalho está em técnicas preventivas.

Prevenção é uma técnica fundamental. Um mecanismo de descoberta requer um mecanismo prevenir modificações no impróprio com rastro de auditoria. Além disso, descoberta no final das contas só é útil à extensão que previne atividade imprópria ameaçando ação punitiva.

Finalmente, há a terceira “técnica" de tolerância, na qual o potencial para algumas brechas de segurança é tolerado; porque estas brechas são muito caras para prevenir ou descobrir, ou a probabilidade da ocorrência deles é considerada baixa, ou medidas de segurança somente são aceitáveis para usuários até um ponto razoável.

Na prática, todo sistema tolera algum grau de risco com respeito a potenciais brechas de segurança. Porém, é importante entender que risco está sendo tolerado e o que está ficando coberto através de mecanismos de prevenção ou detecção (SANDHU,1993).

2.12 SEGURANÇA

Mecanismos de segurança, se preventivo ou de descoberta por natureza, pode ser implementado com vários graus de garantia. A garantia é relacionada diretamente ao grau de esforço exigido do mecanismo. Mecanismos com baixo grau de garantia são fáceis de implantar, mas também, relativamente fácil burlar. Falhas Sutis (bugs), em sistemas ou softwares de aplicação, pode conduzir numerosas brechas de segurança. Por outro lado, mecanismos de altos graus de garantia são estupidamente difíceis de implantar. Eles também tendem a degradar o desempenho das aplicações e sistemas. Com os avanços de desempenhos dos hardwares, isso está compensando a perda de performance destes controles (SANDHU,1993).

2.13 DETERMINAÇÃO DO PROBLEMA

Uma das principais razões que motivam o uso dos SGBD's é o controle centralizado, tanto dos dados, quanto dos programas de acesso a esses dados (SILBERSHATZ,1999).

O processamento do SGBD pode proporcionar grandes benefícios para as organizações, mas, infelizmente, tal processamento também aumenta a vulnerabilidade organizacional. Com um SGBD, os dados da companhia, um recurso valioso, são centralizados ficando prontamente acessíveis. Na verdade, os SGBDs são projetados para maximizar esta acessibilidade. Esta situação é ótima para os usuários autorizados. Mas, infelizmente, os SGBDs também, apresentam facilidades de uso por usuários não-autorizados, e criminosos. Devido a este problema os SGBDs comerciais foram incorporados de características de segurança. Mas, no máximo, estas características só permitem que usuários autorizados (pessoas ou programas) acessem os dados e restringem o tipo de processamento nos dados (KROENKE,1999).

O problema surge quando as exigências de controle de acesso de uma aplicação diferem das políticas possíveis de serem implantas no SGBD. Na maioria dos casos a

política de segurança das aplicações força a uma única solução, a de implantar esta política como parte do código da aplicação. Porém, esta solução torna muito difícil as tarefas de verificação, modificação, e execução adequada das políticas de segurança (BERTINO,1996a).

Em um SGBD, nem todos os seus usuários e programas estão autorizados a ter acesso a todos os dados. Além disso, alguns usuários precisam ter acesso a mais dados do que outros. Alguns precisam de direitos mais amplos de processamento. Existem aqueles usuários que necessitam modificar ou eliminar registros, mas outros usuários não. Da mesma forma, que na grande maioria dos casos determinados usuários devem ter acesso apenas a determinados dados de uma tabela, (registros ou linhas) já outros devem ter acesso a outros registros nesta mesma. Para piorar ainda mais, existem usuários que deve ter acesso de modificação de um determinado conjunto de registros de uma tabela específica, quando estiverem executando uma determinada aplicação, e quando estiverem executando outra aplicação específica deve ter permissão a modificar outro conjunto de registro da mesma tabela. Como atender esta política de segurança?

Em uma análise superficial poderia se dizer, que para solucionar este problema bastaria criar visões de BD para cada usuário, aparentemente simples. Mas em aplicações comerciais desenvolvidos por terceiros, isso não é tão fácil. Principalmente nas de gestão de negócio, os *Enterprise Resource Planning* (ERP's), os quais são utilizados por grandes empresas, onde em alguns casos centenas ou milhares de usuários utilizam o sistema, sendo que este é composto por alguns milhares de programas e tabelas distintas, transformando o processo de liberação do acesso e controle de acesso uma coisa monstruosa e difícil de administrar.

Hoje, com raras as exceções o desenvolvimento de aplicativos tem sua base de dados armazenada em um SGBD relacional. Todos os desenvolvedores de softwares gastam um tempo grande desenvolvendo módulos de controle de acesso para suas aplicações e que geralmente acabam não aderindo às necessidades de segurança das empresas. Como controlar os acesso às informações destes usuários que receberam os privilégios para tal, por outro aplicativo que não tem tal controle?

Com base nestes problemas surge a pergunta, os SGBD estão preparados para centralizar o controle de acesso garantindo as políticas de segurança das empresas, sem a necessidade de mecanismos adicionais?

Caso este questionamento for positivo, será possível diminuir os custos atuais de desenvolvimento de softwares, e principalmente garantir a segurança central dos dados melhorando a administração dos mesmos.

2.14 CONCLUSÃO

Neste capitulo, destaca-se a importância que a segurança de informação assume com um todo nas aplicações das empresas. Dentre estas se destaca a confidencialidade, integridade e a disponibilidade da informação, bem com os tipos de controles de acesso disponíveis.

Para apresentar os controles de acesso das informações descreveu-se a importância que as políticas assumem nos processo de controle.

3 CONTROLE DE ACESSO PADRÃO DO SGBD

3.1 INTRODUÇÃO

Neste capítulo nós discutimos os controles de acesso providos na geração atual de Sistemas de Administração de Banco de dados comercialmente disponíveis, através da linguagem SQL. Nosso foco está em sistemas de relacional.

O controle semântico dos dados disponíveis nos SGBD's é uma das funcionalidades mais importantes dos SGBD's, este inclui gerenciamento de visões, controle de segurança de acesso e controle de integridade semântico e disponibilidade dos dados.

3.2 CONTROLE SEMÂNTICO DE DADOS

Segurança em SGBD refere-se à proteção dos dados contra a divulgação, alteração ou destruição não-autorizada. Segurança garante que os usuários têm permissão para fazer o que estiverem tentando fazer. (DATE,1999), (OZSU,1999) "Uma das principais razões que motivaram o uso de SGBDs é o controle centralizado, tanto dos dados, quanto dos programas de acesso a esses dados". Isso convêm afirmar que o controle centralizado de um SGBD, em um ambiente cliente/servidor, consiste em benefício para a segurança das informações. No entanto, ao fazer uma analogia com os sistemas de informação atuais, observamos que as empresas disponibilizam seus dados a vários tipos de usuários, dentro ou fora do seu ambiente físico (SILBERSHATZ,1999).

Por este motivo o mesmo precisa estar ciente de certas limitações que os usuários não podem violar; essas limitações devem ser especificadas em linguagem apropriada, e mantida no catálogo do sistema ou no dicionário (DATE,1999), (SANDHU,1993).

Nenhum dos sistemas gerenciadores de bancos de dados comerciais fornecem de modo geral, uma capacidade adequada a atender todas as políticas de seguranças que as aplicações e usuários necessitam. Eles apenas apresentam um número restrito de subconjuntos de segurança, (KROENKE,1999), (SANDHU,1993). Para isso podem ser definidas visões, restrições de segurança, e restrições semânticas de integridade como regras que o sistema controla automaticamente. A violação de alguma destas regras por um programa de usuário (um conjunto de operações de banco de dados) geralmente implica na rejeição da ação do programa (OZSU,1999) .

A definição de regras para controlar a manipulação de dados é parte da administração do SGBD. Esta função é executada pelo administrador da base de dados (DBA) e deve atender os requisitos das políticas de segurança definidas pela empresa.

3.2.1 Administrador de banco de dados

É a pessoa responsável por aplicar as políticas organizacionais (OZSU,1999) . É ele quem possui a competência técnica para gerenciar todo o SGBD de uma organização. Suas principais funções envolvem um conjunto de atividades, que partem desde a definição da estrutura e do conteúdo do BD, até atividades relacionadas à administração dos componentes principais do sistema. Dentre esses componentes são mencionados os servidores, estruturas de armazenamento e métodos de acesso, mecanismos de medição de desempenho, cópia de segurança e recuperação de dados, administração de usuários e restrições de integridade (OZSU,1999) .

Outra tarefa importante do DBA é também servir de elo de ligação entre os BDs e os usuários. O DBA define os critérios de autorização através de mecanismos que permitam criar contas de usuário, implementando o critério de segurança apropriado. Além disso, o DBA pode fornecer aos analistas e programadores, todas as informações

necessárias para viabilizar o desenvolvimento de aplicações de BD específicas, que serão utilizadas pelos usuários finais.

Com relação à tecnologia utilizada para resolver as questões de segurança ao nível de usuário, o DBA pode, através do SGBD, implantar mecanismos de segurança baseados em esquemas de login e senha, e também em níveis de privilégios.

Os mecanismos de controle de acesso baseados em login e senha, permitem aos SGBDs, não só garantir ou restringir o acesso dos usuários, como também, registrar todas as operações a partir do período em que o usuário acessa o BD até o momento em que ele encerra suas atividades. Como o foco do trabalho não é o mecanismo de autenticação não será abordado este processo.

Os níveis de privilégios, por sua vez, permitem restringir o acesso a determinado BD a um grupo restrito de usuários. Uma vez definido qual BD poderá ser acessado, poderemos também definir quais os tipos de operações que poderão ser realizadas pelos usuários sobre seus objetos. Vários produtos de SGBD que utilizam SQL implementam métodos conhecidos como concessão (*Grant)* e revogação (*Revoke)* para manter níveis de privilégios dos usuários.

3.3 PRIVILÉGIO

Uma autorização ou privilégio é uma permissão que define o modo como acessar um determinado objeto; por exemplo, autorização para consultar uma tabela. Autorizações podem ser concedidas para permitir que um usuário em particular possa conectar no BD, criar tabela no próprio escopo, selecionar registros de tabelas de outros escopos, ou executar procedimentos de outros escopos armazenados no banco de dados (HAZEL,2001).

O usuário novo quando criado, não recebe nenhum privilégio. A estes usuários novos, primeiro, deve ser concedido o privilégio para efetuar o logon ou executar qualquer operação de banco de dados. Os usuários não podem fazer nada a menos que lhes fossem determinados o privilégio específico para fazer assim. Há um número

grande de privilégios que podem ser dados aos usuários. Existem dois tipos de privilégios disponíveis para ser concedidos. Eles são de sistemas e privilégios de objeto.

3.3.1 Privilégios de sistema

Privilégios de sistemas permitem um usuário criar ou manipular objetos, mas não dá acesso a objetos de banco de dados atuais. Como por exemplo: Alterar tabela, criar tabela, executar qualquer procedimento, e apagar tabela.

Autorizações de sistemas permitem um usuário executar totalmente uma ação particular de sistema ou uma ação particular em um tipo particular de objeto do escopo. Privilégios de sistemas permitem um usuário executar comandos como: criar tabela, alterar tabela, apagar tabelas, remover registros de qualquer tabela do BD executar qualquer stored procedure. Muitas autorizações de sistemas são concedidas somente para administradores e desenvolvedores de aplicações, porque são privilégios muito poderosos (HAZEL,2001).

3.3.2 Privilégios de objeto

São usados privilégios de objeto para permitir acesso a um objeto de banco de dados específico, como uma tabela particular ou visão. Privilégios de objeto que são dados ao nível de visão são especialmente para restringir o acesso a colunas e registros de uma tabela. Este permite um administrador dar aos usuários acesso a um grupo escolhido de colunas ou registros de uma tabela, em lugar de a tabela inteira. É possível também, conceder ao usuário o privilégio para ele repassar o privilégio para outros usuários ou papéis.

Privilégios de objeto permite o usuário executar uma ação particular em um objeto específico do esquema. Por exemplo, o privilégio para deletar registros de uma tabela específica é um privilégio de objeto.

Privilégios de objeto permite a segurança de tabelas ao nível de operações com DML (*Data manipulation language)* e DDL (*Data dictionary language)*. Por exemplo, um administrador pode conceder a um usuário individual o privilégio para usar operações de *DML* seleção*,* inserção*,* modificação, remoção, (*select, insert, delete e update)* em tabela ou visão, ou privilégios de executar operações *DDL* índices*,* alteração, recurso, eliminar, referenciar (*index, alter, resource, drop e references)* em uma tabela.

- **Autorização seleção (select)** – permite a leitura, mas não a modificação, dos dados.
- **Autorização inserção (Insert)** – permite a inserção de novos dados, mas não a modificação de dados existentes.
- **Autorização de modificação (update)** – permite a modificação, mas não a remoção, de dados.
- **Autorização de remoção (delete)** – permite a remoção de dados.

Um usuário pode receber todos, nenhum ou uma combinação desses tipos de autorizações, (SILBERSHATZ,1999). Podemos também conceder direitos para um determinado usuário poder modificar esquemas do banco de dados, estes direitos são:

- **Autorização índices (index)** – permite a criação de índices.
- **Autorização recurso (resource**) – permite a criação de novas tabelas.
- **Autorização alteração (alter)** – permite a adição ou remoção de atributos em uma tabela.
- **Autorização eliminar (drop)** – permite a remoção de tabelas.
- **Autorização referencia (references)** – permite a criação de uma chave estrangeira em uma tabela referenciando uma tabela de outro esquema ou dono.

Existe uma diferença grande entre o direito de eliminar e o de remoção, onde o de remoção apenas remove o registro de tabelas, podendo até remover todas os registros, deixando a tabela vazia. Já o de eliminar, este destrói a tabela tirando a mesma, inclusive, do dicionário de dados de SGBD.

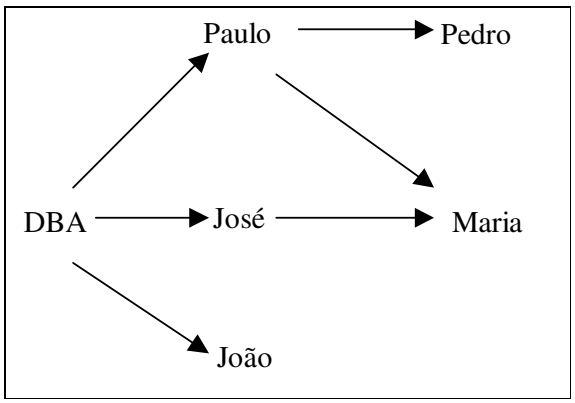
Alguns privilégios podem ser específicos ao nível de colunas. O privilégio de inserção e modificação de tabela possibilita restringir um usuário a colunas individuais de tabelas.

3.3.3 Concessão de privilégios

Podemos permitir que um determinado usuário conceda a outro usuário algum privilégio ou autorização a ele passada. Mas, devemos ter o cuidado ao repassar tal privilégio, este deve ser feito de uma forma que seja possível no futuro ser revogado (SILBERSHATZ,1999).

Ao conceder um privilégio a um usuário, está se montando um grafo de autorizações, com isso, podemos representar a passagem de autorização de um usuário para outro $B_i \rightarrow B_j$. Para esta representação considere a concessão de autorização de seleção na tabela de empregado do banco de dados do sistema de recursos humanos de uma empresa. A concessão de privilégios iniciando pelo administrador de banco de dados o qual concede o privilégio de seleção sobre empregados para os usuários B_1, B_2 e B_3, que pode por sua vez, repassar essa autorização para outros usuários.

Figura 1- Grafo de concessão de autorização



Os nomes, do grafo acima, representam os usuários, os quais são concedidos os privilégios. A cada nova concessão feita a um usuário este é incluído no grafo. A base do grafo é sempre o DBA ou o dono da tabela. Observe que no grafo da figura 1, que

tanto o usuário Paulo como o José concedem privilégios para a usuária Maria, e somente Paulo concede para Pedro.

Um determinado usuário tem um privilégio se, e somente se, houver um caminho a partir da raiz do grafo de autorização até este usuário.

Caso o administrador do banco de dados decida revogar o privilégio concedido ao usuário Paulo, como Pedro recebeu o privilégio de Paulo, seu privilégio deverá ser revogado também. No entanto, Maria teve o privilégio concedido por Paulo, bem como, por José. Como o administrador do banco não revogou o privilégio de repassar o privilégio de José, Maria mantém o privilégio sobre a tabela. Se José revogar o privilégio de Maria, neste caso a mesma perderá o privilégio.

Usuários sem o conhecimento ou mal-intencionados podem tentar anular as regras de revogação de privilégios concedendo privilégios entre si, apresentado na figura 2. Se o administrador revogar o privilégio de José, José pode manter o privilégio por meio de João, como mostra na figura 3, se o privilégio de João for revogado em seguida, João poderia manter o privilégio por meio de José como mostra na figura 4. Entretanto, quando o administrador do banco de dados revoga o privilégio do João, as setas delimitadoras de João para José e de José para João não fazem mais parte do caminho que se inicia no administrador do banco de dados. O banco de dados exige que todas as setas do grafo de autorização faça parte de algum caminho originado do administrador do banco de dados. Essas setas são removidas, conforme podemos observar na figura 5.

Figura 2 - Concessão de privilégios entre si.

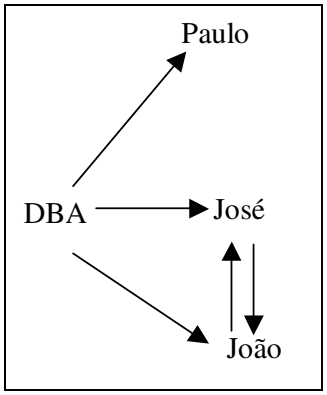


Figura 3- Manter privilégio por receber de outro

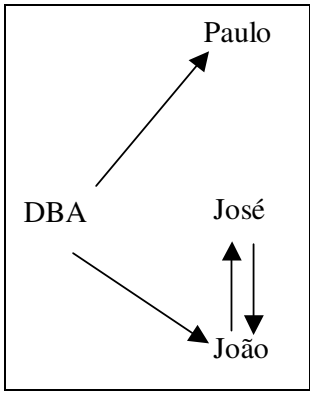


Figura 4 - Tentativa de burlar o controle de segurança.

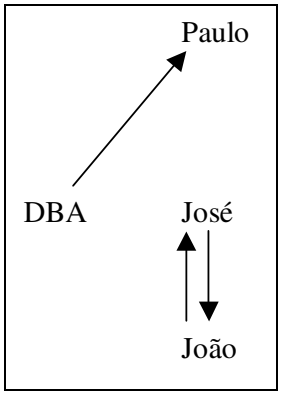
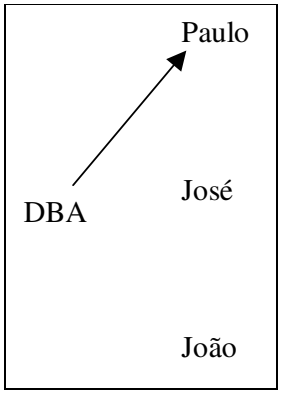


Figura 5 - Grafo de autorização.



3.4 MODELOS DE CONTROLE DE ACESSO E GRANULAÇÃO

Podem ser impostos controles de acesso a vários graus de granulação em um sistema. Algumas possibilidades são enumeradas abaixo.

- Ao banco de dados inteiro.
- Alguns conjuntos de tabelas.
- Uma tabela.
- Algumas colunas de uma tabela.
- Alguns registros de uma tabela.
- Algumas colunas de alguns registros de uma tabela.

Os controles de acesso também, são diferenciados em relação à operação para a qual eles aplicam. Estas distinções são importantes, por exemplo, cada empregado pode ser autorizado a ler o próprio salário, mas não atualizá-lo. Em bancos de dados relacional os modos de controles de acesso são expressados em termos das operações básicas de SQL (como SELECIONE, MODIFIQUE, INSIRA e ELIMINE), como segue.

A habilidade para INSERIR e ELIMINAR é especificada em uma tabela através das tabelas básicas.

SELECIONE é geralmente especificada em uma tabela através das tabelas básicas. Uma melhor granulação de autorização para SELECIONE pode ser provido por visões.

MODIFIQUE, pode ser restringido a certas colunas de uma tabela (SANDHU,1993), (SANDHU,1994).

3.4.1 Controle de acesso dependendo dos dados

Os controles de acesso de banco de dados freqüentemente dependente de dados. Por exemplo, alguns usuários podem ser limitados a ver salários que são menos que R$

3.000,00. Igualmente, um gerente pode ser restringido a ver salários dos empregados do seu departamento. Discutiremos duas técnicas básicas: controle de acesso baseado em visões e consultas modificadas, para implantar controles de acesso dependentes dos dados, em bancos de dados de relacional (SANDHU,1993), (SANDHU,1994).

3.4.2 Controle de acesso baseado em visões

Uma tabela básica é uma tabela real do banco de dados, a qual é armazenada de fato no Banco de dados. Uma visão é uma tabela virtual que é derivada de tabelas básicas e outras visões. O banco de dados armazena as definições das visões e materializa a visão quando necessário (SANDHU,1993).

Abaixo ilustrar o conceito de uma visão, e sua aplicação de segurança, considere a tabela de EMPREGADO de Tabela-1. O seguinte comando SQL define uma visão chamada de DEPARTAMEMTO_20.

```
Create view departamento_20 as
select nome, salario, chefe
from empregado where coddepto = 20;
```

Quadro 1 - Tabela básica de empregados.

NOME	CODDEPTO	SALARIO	CHEFE
SMITH	20	800	7902
ALLEN	30	1600	7698
WARD	30	1250	7698
JONES	20	2975	7839
MARTIN	30	1250	7698
BLAKE	30	2850	7839
CLARK	10	2450	7839
SCOTT	20	3000	7566
KING	10	5000	null
TURNER	30	1500	7698

Quadro 2 - Visão departamento_20

NOME	SALARIO	CHEFE
SMITH	800	7902
JONES	2975	7839
SCOTT	3000	7566

Quando o comando de criação da visão é executado, o comando de seleção que define como deriva a visão, este não é executado, apenas é armazenado no catálogo.

Mas, para o usuário, é como se realmente houvesse uma tabela no banco de dados, chamada de departamento_20, com linhas e colunas com podemos ver na figura-6 abaixo.

Figura 6 - Descrição da visão departamento_20.

```
SQL> desc departamento_20
 Nome                                      Nulo?    Tipo
 ----------------------------------------- -------- -------------
 NOME                                                VARCHAR2(10)
 SALARIO                                             NUMBER(7,2)
 CHEFE                                               NUMBER(4)
```

Esta definição, tabela virtual, é mostrada na tabela-2. Um usuário recebeu acesso para ler departamento_20, este está limitado receber informação sobre empregados do departamento de 10. Isso ilustrar o aspecto dinâmico de visões, suponha que um ADAMS é um novo de empregado, e é inserido na tabela de EMPREGADO, como mostrado em tabela-3. A visão departamento_20 será automaticamente modificada para incluir ADAMS, como mostrado em tabela-4.

Também, podem ser usadas visões para prover acesso à informação estatística. Por exemplo, a seguinte visão dá o salário médio para cada departamento.

```
Create view mediasal(depto,avgsal) as
Select coddepto,avg(salario)
From empregado
Group by copdepto;
```

Para um usuário fazer uma consulta não necessita distinguir entre visão e tabela básica. Uma visão é considerada, simplesmente, uma tabela do SGBD, estas tabelas virtuais são modificadas automaticamente pelo SGBD, quando a tabela ou tabelas básicas são modificadas. As visões provêem um mecanismo muito poderoso para especificar autorização dependente de dados para recuperação de dados. Porém, pode haver problemas de significativos, se visões forem modificadas diretamente pelos usuários (ao invés de efetuar modificações nas tabelas base). Isso ocorre, devido a nossa inabilidade teórica, em geral, traduzir atualizações em visões, em atualizações nas tabelas base. Isto limita a utilização das visões para os controles de acesso baseados em dados para operações de atualização.

Quadro 3 - Modificação da tabela base empregado.

NOME	CODDEPTO	SALARIO	CHEFE
SMITH	20	800	7902
ALLEN	30	1600	7698
WARD	30	1250	7698
JONES	20	2975	7839
MARTIN	30	1250	7698
BLAKE	30	2850	7839
CLARK	10	2450	7839
SCOTT	20	3000	7566
KING	10	5000	
TURNER	30	1500	7698
ADAMS	20	1100	7788

Quadro 4 - Modificação automática da visão departamento_20.

NOME	SALARIO	CHEFE
SMITH	800	7902
JONES	2975	7839
SCOTT	3000	7566
ADAMS	1100	7788

A segurança com visões, se dá pelo fato de podermos criar visões dos dados, de acordo como são às necessidades de restrições que devemos impor para determinado usuário. Podemos restringir a visão dos dados, tanto na horizontal como na vertical, liberando apenas algumas colunas, todas colunas de uma tabela, todas colunas com um número restrito de registros, ou qualquer outra combinação, como podemos também fazer junções de tabelas conforme melhor representar a concessão de acesso. Com isso, o usuário não recebera os privilégios sobre as tabela básica mas, sim sobre as visões.

Como podemos ver acima, o mecanismo de visões do SQL proporciona uma importante medida de segurança. No entanto, a abordagem que se baseia na visão é um tanto inábil, especialmente se algum usuário precisar, ao mesmo tempo, de direitos diferentes sobre subconjuntos diferentes da mesma tabela (DATE,1999).

3.4.3 Consulta modificada

Consulta modifica é outra técnica para garantir controles de acesso dependentes de dados para recuperação. (Esta técnica não é apoiado em SQL, mas é discutido aqui por causa de perfeição.) Nesta técnica, uma consulta submetida por um usuário é modificada para incluir restrições adicionais como determinado pelas concessões feitas ao usuário.

Suponha que o administrador de banco de dados concedeu para o João o privilégio para examinar a tabela básica de EMPREGADO, mas somente para empregados do departamento 20, como segue:

```
Grant select on empregado to João where coddepto = 20;
```

Agora suponhamos que João execute o seguinte comando em SQL.

```
Select nome,coddepto,salário,chege from empregados;
```

Na ausência de controles de acesso esta consulta devolveria a tabela inteira de EMPREGADO. Porém, devido à CONCESSÃO anterior, o SGBD modificará automaticamente esta consulta, conforme segue:

```
Select nome,coddepto,salário,chege from empregados where coddepto = 20;
```

Isto limitará o João a consultar apenas aquela porção da tabela de EMPREGADO para a qual lhe foi concedido acesso de selecionar.

3.5 CONCESSÃO E REVOGAÇÃO DE ACESSO

Em SQL padrão (DATE,2000), a linguagem para especificar a segurança de dados, em base de dados relacionais, propõem características de segurança simplesmente através dos comandos de concessão (grant) e revogação (revoke) de privilégios. O SQL padrão, inclui os privilégios seleção, inserção, modificação e deleção. As instruções de concessão e revogação de privilégios representam a principal interface do usuário para o subsistema de autorização ou privilégios. Estes privilégios são verificados pelo controle de acesso.

O SGBD tem o controle de acesso tipo discriminatório (DAC), controlado em uma única dimensão o acesso a dados. O administrador concede aos usuários privilégios que determinam as operações (como leitura, escrita) que eles podem executar em dados. Para um usuário processar ou executar uma tarefa este deve ter privilégios apropriados para tal, como privilégio de selecionar dados de um determinado objeto, como uma tabela ou visão (SANDHU,1993), (SANDHU,1994).

Desta forma, para que um usuário possa ser capaz de desempenhar qualquer operação em SQL, o mesmo, primeiro deve ter recebido o privilégio para aquela operação; esta operação será rejeitada pela devida mensagem de erro ou código de exceção. Por exemplo, caso o usuário Paulo queira executar a seguinte instrução SQL "Select * from empregados;" de maneira bem sucedida, o mesmo deverá previamente ter recebido o privilégio de seleção na tabela "empregados".

Dentro do processo de concessão e revogação de privilégios, existem tipos de usuários, onde alguns são apenas usuário de banco de dados, que tem uma conta (identificação de usuário e uma senha) para executar comandos sobre as tabelas de outros usuários, os quais são chamados de donos de tabelas (owner). O Conjunto de tabelas de um banco de dados criado em SGBD pertence sempre a um esquema (schema) ou dono. Este usuário, dono de tabelas primeiramente deve ter recebido o privilégio de recurso (resource) para poder criar tabelas e outros objetos no SGBD. Este usuário dono, das tabelas, sobre as mesmas ele tem plenos privilégios e pode repassar inicialmente os direitos para os demais usuários do SGBD para executarem operações sobre as mesmas.

O padrão do SQL especifica que somente o proprietário do esquema pode efetuar alguma modificação nele. Criar ou remover tabelas, adicionar ou retirar atributos de tabelas e adicionar ou retirar índices, somente poderá ser feitos pelo dono do esquema. Mecanismos diferentes destes não são padrões SQL. Segue abaixo os comandos para concessão de privilégios.

A declaração *Grant* é usada para conceder privilégios. A forma básica dessa declaração é a seguinte:

```
Grant <lista de privilégios> on <nome da tabela ou nome da visão> to <lista de usuários>;
```

A lista de privilégios permite à concessão de vários privilégios em um comando. A seguinte declaração concede ao usuário, Paulo, privilégio de seleção na tabela de empregados:

```
Grant select on empregados to Paulo;
```

O privilégio de seleção só pode ser concedido sobre toda tabela, não permitindo darmos o privilégio sobre parte da tabela (horizontal ou vertical), quando queremos controlar o acesso aos dados de uma tabela, desta forma, devemos utilizar visões.

Já o privilégio de modificação pode ser concedido a todos os atributos da tabela com apenas um comando. Quando concedermos o privilégio de modificação em uma declaração de concessão, a lista de atributos sobre os quais o privilégio de modificação é concedido aparece. Logicamente entre parênteses, imediatamente após a palavra-chave update. Caso a lista de atributos for omitida, o privilégio de modificação será concedido sobre todos os atributos da tabela.

O comando próximo concede o privilégio de modificação no atributo salário na tabela de empregados para José.

```
Grant update(salário) on empregados to José;
```

O privilégio de inserção pode especificar, também uma lista de atributos, da mesma forma, como o privilégio de modificação, caso for omitido a lista de atributos, o privilégio de inserção será sobre todos os atributos da tabela. Quando concedermos privilégios restritos a alguns atributos da tabela, o comando de inserção executado pelo usuário que recebeu o privilégio deve se referir apenas a estes atributos, e cada um dos atributos restante da tabela receberão valores padrões, caso for definido um padrão para este atributo no momento da criação da tabela, ou serão ajustados para nulo (null). Quando o usuário não tiver o privilégio de inserção em todos os atributos da tabela, e um dos campos no qual não se tem o privilégio, não tiver padrão (default) e não aceitar nulos, o comando irá falhar, mas não por falta de privilégios, mas sim, devido a tentativa de violação de integridade. Segue comando concedendo privilégio de inserção na tabela de empregados, menos no campo salário.

```
Grant insert(código, nome, data_nascimento, endereço) on empregados to Jose;
```

O privilégio de eliminação de registros de uma tabela é igual ao de seleção, só pode ser concedido sobre toda a tabela. Segue comando concedendo privilégio de eliminação de registros da tabela empregados por Paulo.

```
Grant delete on empregados to Paulo;
```

Quando queremos conceder privilégios a todos os usuários que estão cadastrados no SGBD, e os que virão a ser cadastrados, podemos utilizar uma palavra chave *public*.

Deve-se lembrar que no comando de concessão de privilégios pode-se conceder o privilégio de repassar este para outro usuário. Desta forma, pode ser exemplificada da seguinte maneira: o usuário Paulo recebe o privilégio de seleção e juntamente o direito de conceder esta para outro usuário. Como também, o Paulo pode conceder para João o direito de selecionar juntamente como o direito de repassar, e este pode sucessivamente ir concedendo para os demais usuários, conforme apresentado na seção concessão de privilégios (DATE,1999). Por exemplo:

```
DBA: Grant select on empregados to Paulo with grant option;
Paulo: Grant select on empregados to Joao with grant option;
João: Grant select on empregados to Juca with grant option;
```

E, assim por diante.

Temos ainda os privilégios de referência, alteração, criação de índice e eliminação de tabela. Estes privilégios estão disponíveis nos SGBD's, mas raramente são usados. Apenas o de referência é comum, pois quando temos vários bancos de dados e cada um com um dono diferente em um mesmo SGBD, este privilégio se torna imprescindível para manter a integridade dos dados.

Para revogarmos um privilégio, usamos a declaração revoke. Ela tem a forma semelhante ao da concessão:

```
Revoke <lista de privilágios> on <nome da tabela ou nome da visão> from <lista de
usuário> [restrict | cascade];
```

Então para revogarmos os privilégios anteriormente concedidos, escrevemos:

```
Revoke select on empregados from Paulo cascade;
Revoke update(salário) on empregados from José;
Revoke insert(código, nome, data_nascimento, endereço) on empregados from Jose;
Revoke delete on empregados from Paulo;
```

Conforme foi comentado na seção, concessão de privilégios, a revogação de um privilégio concedido a um usuário pode fazer com que outros usuários também percam esse privilégio. Esta característica é chamada de revogação em cascata. A revogação também pode ser especificada com restrito (restric):

```
Revoke select on empregados from Paulo restric;
```

Pode ser revogado apenas o privilégio de concessão de privilégios, mantendo o privilégio efetivo, da seguinte forma:

```
Revoke Grant option for select on empregados from Paulo;
```

3.6 PAPÉIS

Devido ao conceito básico de controle de acesso, baseado em papel (RBAC), permissões são associadas com papéis, e os usuários são feitos sócios de papéis adequados e, assim, adquirem permissões dos papéis. Esta idéia surgiu com o advento de computação multi-usuário (SANDHU,1996), (SANDHU,1997), (SIMON,1997).

A principal característica de uso de papéis é a flexibilidade, que provê para a administração de grandes números de privilégios em objetos, reduzindo o esforço para definir e administrar políticas de segurança complexas. Mas os sistemas disponíveis atualmente, a granulariedade do controle de acesso não são suficientes para satisfazer as exigências de aplicações particulares. Por exemplo, controle de acesso em tabelas de banco de dados relacional não pode ser definido a um subconjunto dos registros específicos, é necessário fazer uso de visões de banco de dados para satisfazer este tipo de exigência (GIURI,1997).

De acordo com (BALDWIN,1990), um papel está definido pelo conceito de domínio de proteção nomeado (NPD). Neste modelo, um papel é uma explícita representação de uma coleção de privilégios que estão definidos e usados pelos administradores de sistemas e usuários. Em um papel podem ser definidos privilégios e inclusive outros papéis. A definição de papel é representada pela sintaxe seguinte:

$$r = role(priv_1, \ldots, priv_n, r_1, \ldots, r_n)$$

Na expressão anterior, **r** representa o nome atribuido ao novo papel, $\mathbf{priv_1, \ldots, priv_n}$, são os privilégios concedidos diretamente ao papel, e $\mathbf{r_1, \ldots, r_n}$, são os nomes de sub-papéis concedidos diretamente a **r**.

Privilégio concedido direto e/ou sub-papel concedido direto podem eventualmente estar ausentes de uma definição de papel. Os sub-papéis de **r** são constituídos por **r** e os sub-papéis de seu sub-papel direto.

Com relação à definição anterior, o domínio de proteção, que corresponde a um papel **r**, está composto de seus privilégios concedidos diretos, e os privilégios de seus sub-papéis, está formalmente definido pela função **PD:**

$$PD(r) = \{priv_1, \ldots, priv_n\}, PD(r_1,), \ldots, PD(r_n)$$

Não são permitidas recursividades na definição de papéis. Isto significa que o jogo de papéis dentro de um sistema tem uma estrutura hierárquica onde à parte de baixo consiste nos papéis que não tem nenhum sub-papel.

Um papel pode ser ativado. Esta ativação correspondente a aplicar o domínio de proteção. Geralmente, podem ser nomeados papéis a usuários que podem ativar tais papéis e os sub-papéis. Por exemplo, os mecanismos do SGBD Oracle modelam o NPD, de acordo com esta política (ORACLE,1992).

Porém, políticas diferentes podem ser adotadas para a tarefa e para ativação de papéis. Por exemplo, um papel poderia ser concedido a uma transação específica, a fim de, restringir o acesso de usuário à um pouco de informação (ativação específica de um domínio de proteção) só quando ele executar uma operação em particular.

Em um sistema baseado em papéis (SIMON,1997), o administrador controla o acesso, construindo políticas de segurança e nomeando aos usuários, o que definimos com papéis. Para definir um papel pressupõe definir três grupos:

- Grupo de operações ou ações, isso inclui o que pode ser feito no papel.
- Grupo de objetos que o papel terá acesso.
- Grupo de usuários ou papéis que farão parte deste papel específico.

Para poder criar o papel no SGBD o usuário deve ter o privilégio para tal, “Grant create role to <usuário>;” de posse deste, pode executar o seguinte comando SQL para cria o papel:

```
Create role <nome do papel> [identified by <senha>];
```

Exemplos:

```
Create role lancamento_contabil;
Create role libera_pagamento identified by senhadificil;
Create role verifica_financiamentos identified by senha_dificil;
```

Com os comandos, acima, criamos os papéis, mas até agora não foi concedido nenhum privilégio de executar operações em tabelas, como também, a estes papéis não foram associados a nenhum usuário. Para que isso ocorra é necessário que seja concedidos direitos a estes papéis. Os comandos para esta concessão são os mesmo usados para concessão de privilégios diretamente a usuários, conforme segue:

```
Grant select, insert, delete, update on lancamentos to lancamentos_contabeis;
Grant insert, delete, update on titulos to libera_pagamento;
Grant select on titulos_financiados to verifica_financiamentos;
```

Durante à concessão dos privilégios estamos definindo os dois primeiros grupos dos papéis, os grupos de operações e os grupos de objetos. Para definirmos os grupos de usuários, ou outros papéis, temos que conceder os papéis aos usuário ou papéis, conforme segue:

```
Grant lancamentos_contábeis to Paulo;
Grant titulos_financiados to libera_pagamento;
Grant libera_pagamento to Jose;
Grant lancamentos_contábeis, libera_pagamento to Pedro;
```

Pode-se seletivamente habilitar ou desabilitar os papéis concedido a um usuário. Isto permite controle específico dos privilégios de um usuário em determinada situação. Você pode proteger uso de papéis com uma contra-senha. Podem ser criadas específicas aplicações para habilitar um papel quando for fornecida a contra-senha correta. Os usuários não podem habilitar o papel se eles não souberem a contra-senha.

As seguintes propriedades de papéis permitem administração mais fácil de privilégio:

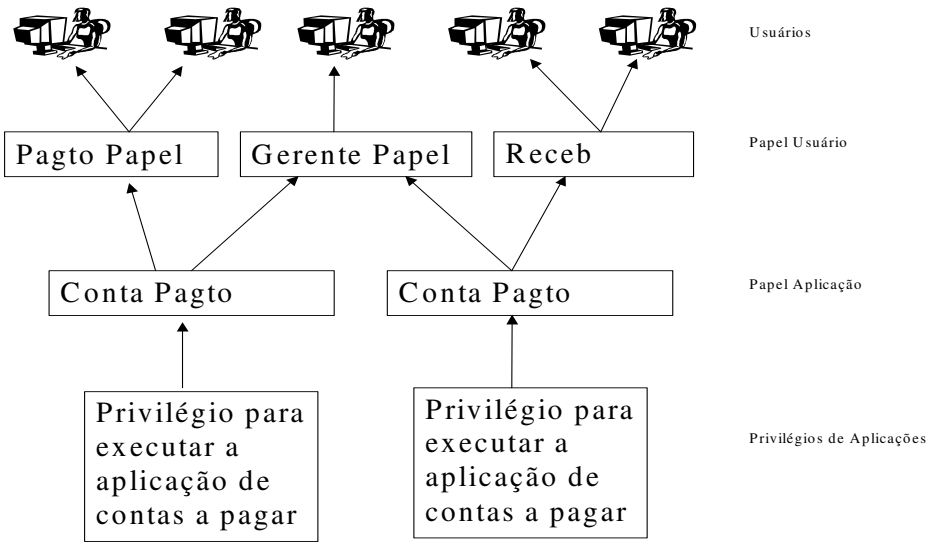
- Reduz a necessidade de concessão de privilégios: Em lugar de conceder o mesmo conjunto de privilégios específico a muitos usuários, o administrador de BD pode conceder os privilégios para papel relacionado

a um grupo de usuários. O DBA pode conceder o papel para cada pessoa do grupo.

- Administração de privilégio dinâmica: Quando os privilégios de um grupo tiverem que mudar, só os privilégios do papel precisam ser modificados. Automaticamente o domínio de segurança de todos os usuários que receberam o papel, sofrem as mudanças feitas ao papel.
- Disponibilidade seletiva de privilégios: Os papéis concedidos a um usuário podem ser seletivamente habilitados (disponível para uso) ou desabilitado (não disponível para uso). Isto permite controle específico dos privilégios de um usuário em determinada situação qualquer.
- Consciência de aplicação: Uma aplicação de BD pode ser projetada para habilitar e desabilitar papéis específicos automaticamente, quando um usuário for usar a aplicação.

Usando vários níveis de papéis e privilégios, você pode alcançar detalhes aumentado controles de acesso, pois quanto menos privilégios melhor, como ilustrado na figura a seguir.

Figura 7 - Exemplo de utilização de papéis



3.7 CONSLUSÃO

Neste capitulo, foram descritos os controles semânticos dos dados encontrados nos SGBDs padrões de mercado, bem como os tipos de privilégios passiveis de serem utilizados com suas respectivas características.

Foram definidos ainda os modelos de controle de acesso e suas granulações, destacando-se os controles de acesso dependendo dos dados, baseados em visões.

Além disso, o processo de concessão e revogação dos privilégios de acesso utilizados pela linguagem SQL, bem como a utilização de papeis como método para facilitar o processo de concessão de privilégios.

4 CONTROLE DE ACESSO EM SGBD ORACLE

4.1 INTRODUÇÃO

O SGBD necessita estar ciente de certas limitações que os usuários não podem violar; essas limitações devem ser especificadas em linguagem apropriada e mantida no catálogo do sistema ou dicionário. O SGBD deve monitorar as interações dos usuários, de alguma forma, para assegurar que as limitações são, de fato, observadas.

Como já foi citado no capítulo anterior todos os SGBDs, inclusive o Oracle tem os padrões de segurança básicos que a tecnologia deve assegurar:

- Confidencialidade.
- Integridade.
- Disponibilidade.

Confidencialidade, integridade e disponibilidade são marcas da segurança do SGBD. Quem pode ter direito de acesso aos dados? Qual porção de dados um usuário poderia acessar? Quais operações poderia um usuário estar autorizado a executar nos dados? Pode estar autorizado o acesso válido aos dados para um usuário quando necessário?

Autorizações são permissões concedidas para um usuário, programa, ou processo para acessar um objeto ou conjunto de objetos. O tipo de acesso aos dados concedido aos usuários pode ser de uma simples leitura ou de leitura e gravação. Estes privilégios

são do tipo DML, operações que o usuário pode executar nos dados armazenados no SGBD.

Neste capítulo serão abordados apenas os itens que fazem referência ao controle de acesso aos dados, os demais tópicos de segurança encontrados nos SGBD Oracle não fazem parte do escopo desta dissertação.

4.2 PORQUE UTILIZA ORACLE PARA PESQUISA

Optou-se em utilizar o SGBD Oracle com parte da pesquisa, por ele ser o mais utilizado no mercado conforme pesquisas realizadas pelo Gartner Group e divulgada pela Globo News.

> "A Oracle lidera o mercado mundial, sob o critério de novas licenças vendidas em 2000, segundo o Gartner Group. Detém 33,8% de participação, seguida da IBM (30,1%) e da Microsoft (14,9%). A Informix vem em quinto lugar, com 3%. Somadas, IBM e Informix praticamente embolam o primeiro lugar junto com a Oracle." (GLOBONEWS,2001)

> "A Oracle Corporation anuncia a sua mais nova versão de banco de dados, o Oracle *9i*. O Oracle9*i* Application Server (servidor de aplicações) já está disponível ao mercado brasileiro desde janeiro deste ano. Segundo pesquisa mais recente do instituto Dataquest (unidade do Gartner Group), a Oracle domina esse mercado, que movimenta US$ 8,8 bilhões. O relatório tem como base estimativas de faturamento com novas licenças de bancos de dados no ano 2000." (PROCESSOR,2001)

4.3 PRIVILÉGIOS DE SISTEMA E DE OBJETO

Um privilégio é permissão para ter acesso a um objeto nomeado de uma maneira prescrita; por exemplo, permissão para examinar uma tabela. São concedidos privilégios a usuários à discrição de outros usuários (os administradores). Podem ser concedidos privilégios para permitir um determinado usuário de conectar-se ao banco de dados (create session); criar uma tabela no próprio esquema; selecionar registros de uma tabela de outra pessoa; ou executa um stored procedure de outra pessoa.

As seções seguintes descrevem as duas categorias distintas de privilégios dentro de um banco de dados:

- Privilégios de sistemas.
- Privilégio de objeto do esquema.
- Manter privilégios de sistema e de objetos.

4.3.1 Privilégio de sistema

Privilégio de sistema, permite um usuário executar uma ação de sistema ou uma ação qualquer, em um tipo particular de objeto do esquema. Por exemplo, o privilégio para criar um usuário, ou para eliminar registros de qualquer tabela do BD, são privilégios de sistema. Muitos privilégios de sistemas são disponíveis somente para administradores e desenvolvedores de aplicações, porque são privilégios muito poderosos (LEVINGE,2002a).

4.3.2 Privilégio de objeto do esquema

O acesso aos dados é geralmente controlado ao nível de acesso do próprio SGBD, ou por tabela específica. Privilégio de objeto do esquema, permite o usuário executar uma ação particular em um objeto específico do esquema. Por exemplo, o privilégio para eliminar registros de uma tabela específica é um privilégio de objeto.

Privilégio de objeto do esquema por tabela, permite: a segurança de tabelas ao nível de linguagem de manipulação de dados (DML); operações de linguagem de dicionário de dados, ou linguagem de definição de dados (DDL). Por exemplo, um administrador pode conceder a um usuário o privilégio para usar operações de DML, eliminação, inserção, seleção e modificação em tabela ou visão, ou privilégios de executar operações DDL alteração, indexação e referência em uma tabela.

Privilégios podem ser específicos ao nível de colunas. O privilégio de inserção e de modificação de tabela, possibilita restringir à colunas específicas da tabela para um usuário. Da mesma forma, privilégios podem ser específicos ao nível de registros, estes restringem o usuário executar seleção, inserção, modificação e eliminação de registros específicos de tabelas.

Como regra geral, privilégios de objetos só podem ser concedidos pelo dono do objeto. Porém, um dono também pode especificar que um usuário particular tem o direito para conceder um privilégio a outros. O range de todos os privilégios para alguma ação em algum objeto do esquema são geralmente concedidos como padrão para o administrador. Em particular, um administrador pode conceder para um programador de aplicação ou para um DBA, o privilégio para conceder qualquer privilégio de objeto "*GRANT ANY OBJECT PRIVILEGE*". Tendo este privilégio, para o desenvolvedor poder ficar mais fácil a tarefa de configuração de segurança que ele enfrenta, e com isso, pode ajudar o DBA solucionando problemas de controle de acesso como surgem (LEVINGE,2002a).

4.3.3 Manter privilégios de sistema e de objetos

A habilidade do usuário em prover um ID e contra-senha válida, pode ser usada como um primeiro nível de autorização para este usuário ter acesso um BD ou tabelas específicas de um BD. Várias técnicas adicionais podem auxiliar a gerência dos privilégios de sistema e privilégios de objetos:

- Usando papéis para administrar privilégios.

- Usando stored procedures para administrar privilégios.
- Usando instalações de rede para administrar privilégios.
- Usando visões para administrar privilégios.

4.3.4 Usando papéis para administrar privilégios

Um papel pode ser usado como mecanismo para prover autorizações, para uma pessoa ou grupo de pessoas pode ser concedido um papel ou um grupo de papéis. Definindo diferentes tipos de papéis para os usuários o administradores pode controlar mais facilmente os privilégios. Existem alguns níveis de papéis;

- Papéis de banco de dados.
- Papéis globais.
- Papéis de empreendimento.
- Papéis de Aplicação seguros.

4.3.4.1 Papéis de banco de dados

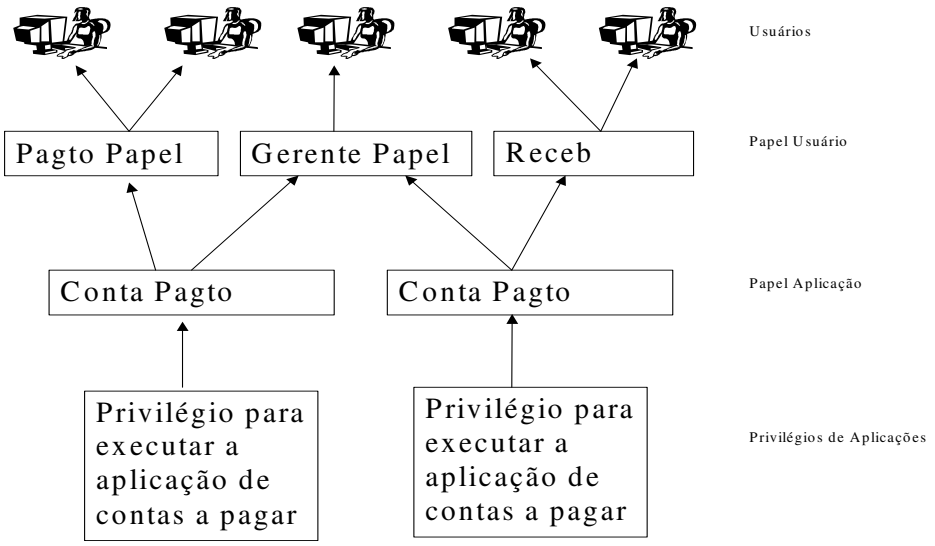
Privilégios permitem aos usuários ter acesso e modificar os dados no BD. São concedidos a papéis de banco de dados, grupos de privilégios relativos a uma função de trabalho específica, que é concedida a usuários ou outros papéis, porque com o uso de papéis a administração dos privilégios fica fácil e melhor. O normal é conceder privilégios ao papel e não para usuários específicos. Pode-se seletivamente habilitar ou desabilitar os papéis concedidos a um usuário, permitindo, assim, controle específico dos privilégios de um usuário em determinada situação qualquer podendo ocorrer proteção do uso de papel, com uma contra-senha. Especificamente podem ser criadas aplicações para habilitar um papel quando for fornecida a contra-senha correta. Os usuários não podem habilitar o papel se eles não souberem a contra-senha (LEVINGE,2002a).

As seguintes propriedades de papéis permitem administração de privilégio mais fácil:

- Reduz a necessidade de concessão de privilégios: Em lugar de conceder o mesmo conjunto de privilégios específicos a muitos usuários, o administrador de BD pode conceder os privilégios para um papel relacionado a um grupo de usuários. O DBA pode conceder o papel para cada pessoa do grupo.
- Administração de privilégio dinâmica: Quando os privilégios de um grupo tiverem que mudar, só os privilégios do papel precisam ser modificados. Automaticamente o domínio de segurança de todos os usuários que receberam o papel, sofrem as mudanças feitas ao papel.
- Disponibilidade seletiva de privilégios: Os papéis concedidos a um usuário podem ser seletivamente habilitados (disponível para uso) ou desabilitados (não disponível para uso). Isto permite controle específico dos privilégios de um usuário em determinada situação qualquer.
- Consciência de aplicação: Uma aplicação de BD pode ser projetada para habilitar e desabilitar papéis específicos, automaticamente, quando um usuário for usar a aplicação.

Usando vários níveis de papéis e privilégios, pode-se alcançar aumento no nível detalhe de controles de acesso, pois quanto menos privilégio, melhor, como ilustrado na figura seguinte:

Figura 8 - Uso comum para papéis



4.3.4.2 Papéis globais

Papéis globais são componentes de segurança de usuário de empreendimento. Um papel global aplica-se somente para um BD, mas pode ser concedido a um Papel de empreendimento, definido no diretório de empreendimento. Embora, um papel global possa ser administrado em um diretório, seus privilégios são contidos dentro de um único BD para o qual estão definidos os direitos.

Define-se o papel global localmente no BD, concedendo privilégios e papéis ao mesmo, mas não pode conceder o papel global para qualquer usuário ou para qualquer outro papel no BD. Quando um usuário do empreendimento tentar conectar ao BD, o diretório de empreendimento é examinado para verificar quais papéis globais estão associadas a este usuário (LEVINGE,2002a).

4.3.4.3 Papéis de empreendimento

Um papel de empreendimento é uma estrutura de diretório que pode conter papéis globais em múltiplos BDs, e que pode ser concedido a usuários de empreendimento.

Armazenando e administrando papéis de empreendimento em um serviço de diretório (LDAP- based), pode-se centralizar a administração de informação relacionada a um usuário, inclusive autorizações.

Por exemplo, o papel de empreendimento *SECRETARIO* poderia conter o Papel global RHSECRETARIO com seus privilégios únicos no BD de recursos humanos, e o papel de ANALISTA com seus privilégios únicos no BD de Folha de pagamento.

Um papel de empreendimento pode ser concedido, ou pode ser revogado de um ou mais usuários de empreendimento. Por exemplo, poderia se conceder o *SECRETARIO* de papel de empreendimento a vários usuários de empreendimento mantendo o mesmo trabalho. Esta informação é protegida no diretório e só, o administrador, pode administrar os usuários e concessão e pode revogar os papéis (LEVINGE,2002a).

A um usuário, podem ser concedidos papéis locais, e privilégios em um BD, além de, papéis de empreendimento.

4.3.4.4 Papéis de aplicações seguras

Há muito tempo existe um problema de segurança, como limitar os usuários terem acesso aos dados, prevenindo que os usuários contornem a lógica das aplicações para ter acesso diretamente aos dados. Por exemplo, em aplicações baseadas em Web, até mesmos usuários conhecidos pelo BD, pode-se não desejar permitir que estes tenham acesso direto aos dados. Até hoje, este é um problema de segurança muito difícil de se resolver, porque não se encontrou nenhum modo seguro para validar a aplicação que é usada para ter acesso aos dados. Por exemplo, um usuário malicioso poderia escrever um programa que parece ser uma aplicação de recursos humanos aparentemente válida.

Um modo para auxiliar este desafio é com o uso de papel de aplicação seguro: um papel implementado por um pacote. O pacote pode executar qualquer validação e assegurar que às condições apropriadas sejam conhecidas antes que o usuário possa executar os privilégios concedidos ao papel no banco de dados. O banco de dados

assegura que somente pacote confiado implementa o papel que determina às condições de acesso corretas.

Um papel de aplicação seguro que é usado por uma aplicação, só pode ser habilitado pela aplicação, e não precisa de uma contra-senha.

Esta característica do Oracle permite difundir o uso de papéis em critérios definidos por usuário. Por exemplo, poderia escrever um papel que permite o uso de um determinado papel por um usuário que só conecta de um endereço determinado IP, ou só tendo acesso o BD por uma particular de camada intermediária.

Em sistemas de três-camadas que usam autenticação por procuração, o pacote pode validar que usuário criou a sessão por uma camada intermediária, e é, desta forma, que o usuário pode acessar o DB por uma aplicação correta. O papel de aplicação seguro, também pode assegurar que um usuário que conecta diretamente ao BD não possa ter acesso a qualquer dado. Um papel de aplicação seguro pode forçar outras condições de segurança, bem como; por exemplo, o usuário pode não estar permitido ter acesso aos dados de recursos humanos, especialmente sensíveis para Internet (LEVINGE,2002a).

Um papel de aplicação seguro eleva a forte verificação nativa, dá controle de acesso fino e granulado do banco de dados para impedir que os usuários assumam qualquer privilégio, a menos que, as condições de acesso corretas sejam conhecidas. O papel de aplicação seguro resolve um assunto de segurança muito difícil, o de dar suporte seguro de acesso aos dados de aplicação baseada na Web.

4.3.5 Usando stored procedures para administrar privilégios

Por stored procedures pode-se restringir às operações que os usuários podem executar em um BD. Permite-se somente terem acesso aos dados por procedimentos e funções que executam com os privilégios pré-definidos. Por exemplo, pode-se conceder aos usuários de terem acesso a um procedimento que atualiza uma tabela, mas negar acesso diretamente a tabela. Quando um usuário invocar o procedimento, este executa

com os privilégios do dono do procedimento. Usuários que têm só o privilégio para executar o procedimento (mas não os privilégios para examinar, modificar, ou eliminar da base das tabelas) podem invocar o procedimento, mas eles não podem manipular dados de tabela de qualquer outro modo (LEVINGE,2002a).

4.3.6 Usando instalações de rede para administrar privilégios

Papéis de banco de dados podem potencialmente ser traçados para serviços externos (como grupos de DCE e autorizações de RADIUS), de forma que, possam administrar centralmente e administrar privilégios para todos os recursos de rede, sendo, que bancos de dados são só um pedaço (LEVINGE,2002a).

4.3.7 Usando visões para administrar privilégios

Em lugar de conceder privilégios de usuários em uma tabela particular, você pode lhes dar acesso para uma visão da tabela. Visões adicionam mais dois níveis de segurança:

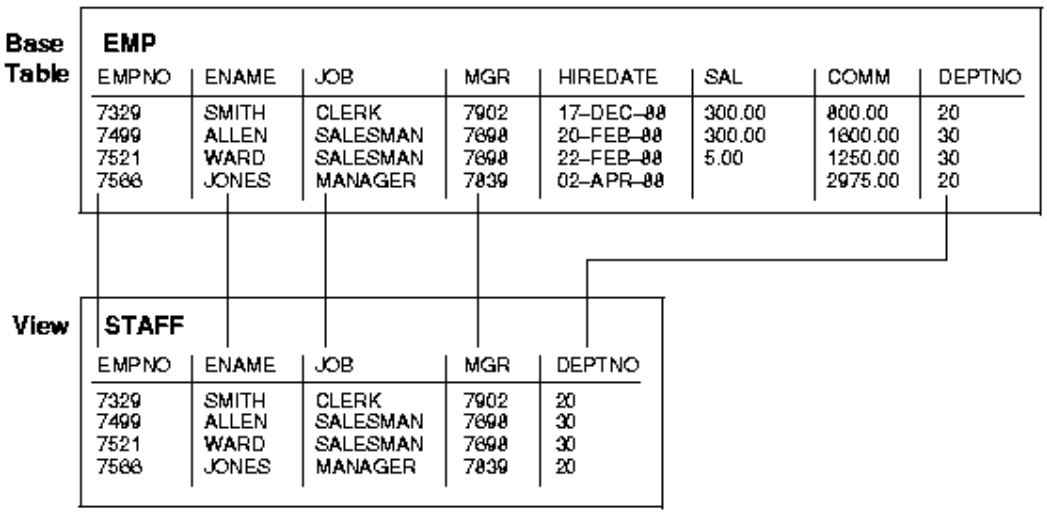
Uma visão pode limitar acesso a colunas somente selecionadas da Tabela base.

Uma visão pode prover segurança baseada em valor da informação em uma tabela. Assim, uma cláusula *"*onde*",* na definição de uma visão, pode exibir registros somente selecionados de uma tabela base.

Usar uma visão só requer privilégios apropriados para a própria visão. Não há necessidade de dar privilégios em objetos de base da visão para usuários, basta dar direitos na visão.

O exemplo a seguir é de uma visão chamado STAFF derivado da tabela base EMP. Note que a visão mostra apenas cinco das colunas da tabela básica.

Figura 9 - Exemplo de uma visão.



Fonte: Oracle9i Security Overview

Visões podem restringir acesso de usuário a um jogo predeterminado de registros e colunas de uma tabela.

4.4 SEGURANÇA AO NÍVEL DE REGISTRO

Uma forma mais granular de acesso de dados é o acesso ao nível de registro. Para uma tabela qualquer com dados, ter acesso a registros específicos pode estar baseado em algumas considerações, como o departamento para o qual os empregados pertencem, a responsabilidade no trabalho deles, ou a sua titulação, ou outros fatores significantes. No passado, foram usadas visões complexas e dinâmicas para implantar segurança ao nível de registro. Porém, há duas abordagens mais efetivas a este problema: Banco de dados privado virtual (VPD), no qual você cria sua própria implementação de segurança ao nível de registro; e controle de acesso rótulo-baseado (label-based), no qual você personaliza uma política de VPD, a qual já é feito para realizar isto. Esta seção descreve estas abordagens alternativas (LEVINGE,2002a).

- Visões complexas e Dinâmicas.
- Reescrever consultas de aplicação: Banco de dados Privado Virtual.

- Controle de acesso baseado em rótulo.

4.4.1 Visões complexas e dinâmicas

Visões complexas e visões dinâmicas estão entre as abordagens históricas de segurança ao nível de registro. As definições de visão complexas é o resultado de quando programadores constróem suas próprias tabelas de segurança de usuários e juntam as tabelas da aplicação com as novas tabelas de segurança de usuário, baseada no nome do usuário da aplicação. Esta abordagem, geralmente, requer muitas definições de visão complexas, nas quais devem ser feitas manutenções com a mudança de exigências de segurança. Outra abordagem é a criação de visão dinâmica. Esta abordagem usa execuções dinâmicas de utilidades em DDL para criar uma nova definição de visão, baseado na identidade do usuário da aplicação. Porém, usando visões dinâmicas, o valor financeiro é elevado e consome muito tempo (LEVINGE,2002a).

4.4.2 Reescrever consultas de aplicação: VPD

Banco de dados privado virtual é uma habilidade para executar modificação de consulta baseado em uma política de segurança que é definida em um pacote, o qual é associado a uma tabela, visão, ou sinônimo. O banco de dados privado virtual disponibiliza um controle de acesso de granulação fina no qual é dirigido a dados, este é dependente do contexto, e baseado em registro. É uma tecnologia que possibilita construir principalmente sistema de três-camadas, que apresentam recursos de missão críticas.

4.4.3 Banco de dados privado virtual em Oracle9i

Oracle9i provê controle de acesso ao nível de registro por seu banco de dados privado virtual (VPD) estando somente disponível em Oracle. Além disso, o Oracle

suporta rótulo de segurança no produto, o qual está embutido no conjunto de ferramentas do banco de dados privado virtual com o controle de acesso baseado em rótulos.

- Banco de dados privado virtual em Oracle8i e Oracle9i.
- Como trabalha o banco de dados privados virtuais.
- Contexto de aplicação em Oracle9i.
- Como contexto de aplicação facilita VPD.
- Como dividir o controle de acesso de granulação fina para facilita VPD.
- Modelo de usuário e banco de dados privado virtual.
- Gerente de política de Oracle.

4.4.3.1 Banco de dados privado virtual em Oracle8i e Oracle9i

Oracle fixou um novo padrão em segurança de banco de dados com a introdução do banco de dados privado virtual (VPD): servidor reforçado, no controle de acesso granulação fina, junto com contexto de aplicação segura, permite que muitos clientes e parceiros tenham acesso direto e seguro os dados de missão crítica. Dentro de um único banco de dados, o banco de dados privado virtual habilita controle de acesso de dados por usuário ou por cliente com a garantia de separação física de dados. Para acesso de Internet, banco de dados privado virtual pode assegurar, por exemplo, que clientes bancários on-line vêem só suas próprias contas. Já Web que é anfitrião de companhias pode manter os dados de várias companhias no mesmo banco de dados, permitindo que cada companhia apenas veja seus próprios dados.

De acordo com (LEVINGE,2002a) utilizando do banco de dados privado virtual podemos centralizar o controle da segurança no SGBD, com isso, as aplicações não necessitam mais ter seus próprios controles de segurança, deixando assim, a segurança dos dados mais forte, porque está centralizada no banco de dados, mantendo a mesma independente de qualquer aplicação que o usuário utilizar para acessar os dados. Com isso, não tem como um usuário tentar burlar a segurança, tentando acessar os dados diretamente ou usando uma ferramenta nova para escrever consultas. O banco de dados

privado virtual é a tecnologia ideal para organizações que disponibilizam serviços de hospedeiro e aplicações baseadas em Web.

O VPD foi incorporado a versão 8i do SGBD Oracle com as seguintes características: controle de acesso de granulação fina, garantindo a segurança ao nível de registro para todas as aplicações, ligando estas políticas de segurança diretamente as tabelas e visões. Na versão 9i do Oracle foram ampliadas algumas características conforme segue:

- Gerente de Política Oracle, uma ferramenta para facilitar administração de política de segurança.
- Divisão do controle de acesso de granulação fina, para facilitar o desenvolvimento e o uso do VPD em varias aplicação e ambientes hospedeiros.
- Contexto global de aplicação, para suportar modelos de aplicação de usuário.
- VPD para sinônimos.

4.4.3.2 Como trabalha o banco de dados privados virtuais

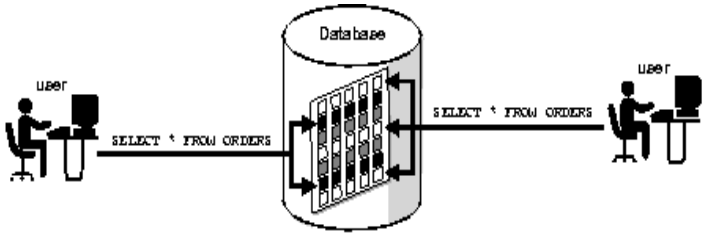
O VPD tem a capacidade de associar uma ou mais políticas de segurança a tabelas ou visões. Uma política de segurança é uma restrição no tipo de acesso ou visão dos dados que um usuário pode adquirir. Para acessar direta ou indiretamente uma tabela com uma política de segurança fixada a ele no banco, é necessário consultar uma função que implementa esta política. Esta função política retorna uma condição de acesso conhecida como um predicado (uma clausula ONDE), o qual o SGBD anexa ao comando SQL do usuário, com isso, modificando dinamicamente o acesso aos dados do usuário.

O VPD pode ser implementado escrevendo um stored procedure para anexar um predicado SQL para cada comando SQL controlando o acesso ao nível de registro para aquele comando. Sua política de segurança então, liga a função ao esquema e tabela desejado. Por exemplo, se João que pertence ao departamento 10 tentar usar o comando

"*select * from emp*", pode se usar o VPD para acrescentar a cláusula "*where depto=10*". Deste modo, a consulta é modificada para restringir o acesso a apenas os registros que o usuário tem permissão de ver.

No contexto de uma aplicação segura, pode-se basear em qualquer atributo que na aplicação julgar significante para garantir a condição de acesso no virtual, como organização, centro de custos, número de conta, ou posição. Por exemplo, em um sistema na Web de pedidos, o acesso pode estar baseado no número do cliente, e se o usuário é cliente ou um representante de vendas. Deste modo, o cliente pode ver como está seus pedidos on-line (mas apenas os seus pedidos), enquanto os representantes de vendas podem ver os vários pedidos, mas só os dos seus clientes os quais ele representa.

Figura 10 - VPD - Cliente pode ver somente suas ordens.



Fonte: Oracle9*i* Security Overview, Release 2 (9.2).

O VPD assegura que, independente da ferramenta ou aplicativo que o usuário usar para acessar os dados, a mesma política de controle acesso será usada fortemente. Deste modo, o VPD pode ajudar assegurar que clientes apenas possam ver suas contas e de nenhum outro, isso facilita muito o dia-a-dia das empresas, por exemplo, de telecomunicações que podem manter os registros do cliente salvos separadamente, facilitando o controle das complexas regras dos sistemas de recursos humanos (LEVINGE,2002a).

Esta capacidade de controle de acesso de granulação fina, também se aplica quando um sinônimo é usado no banco de dados. Função de política aplicada a sinônimo pode criar os mesmos controles que anteriormente eram impostos quando

usávamos visões, mas agora sem os custos em recursos e processamento, o qual cresce proporcionalmente com o número de usuário.

4.4.3.3 Contexto de aplicação em Oracle9i

O controle de acesso de granulação fino é facilitado pelo contexto de aplicação. Isso, permite implantações de políticas de segurança com funções e então associar estas políticas de segurança com as aplicações. Cada aplicação pode possuir seu próprio contexto específico de aplicação. Ao usuário não é permitido trocar arbitrariamente de contexto.

Contextos de aplicações permitem um controle de acesso flexível, baseado em parâmetros, baseado em atributos de interesse da aplicação. Por exemplo, para uma aplicação de recursos humanos o contexto poderia ser a “posição”, “unidade organizacional”, e “pais”, enquanto um atributo de controle de entrada de pedidos poderia ser “número do cliente” e “região de vendas”.

4.4.3.4 Como contexto de aplicação facilita VPD

Na maioria das aplicações contém informações na base, sobre o qual, o acesso será limitado. Em uma aplicação de entrada de pedidos, por exemplo, deveria limitar os clientes acessar apenas seus próprios pedidos (PEDIDO_NUMERO) e número do cliente (CLIENTE_NUMERO). Contexto de aplicação é uma característica fundamental do SGBD, o qual permite definir, grupos, e atributos de acesso, que uma aplicação pode usar para controlar o acesso. Seguramente pode-se armazenar o atributo do usuário como um nome do usuário, número de empregado, e a posição dele na hierarquia de administração. Esta informação pode ser resgatada depois na sessão, e usada para o controle de acesso de granulação fina.

O contexto de aplicação pode ser inicializado de quatro modos diferentes:

- Contexto de aplicação teve acesso localmente:

Dentro de um ambiente local de banco de dados, os valores de atributos podem ser inicializados na sessão de informação do usuário. Cada aplicação pode ter seu próprio contexto com seus próprios atributos.

- Contexto de aplicação inicializada externamente:

Esta característica deixa especificar um tipo especial de espaço, de nome que aceita inicialização de valores de atributos de recursos externos. Isto aumenta o desempenho e facilita a propagação automática do atributo de uma sessão para outra. Algumas aplicações, armazenam atributos usados no controle de acesso de granulação fina, dentro de tabela de metadados, de um banco de dados específico para controlar o acesso. Por exemplo, uma tabela de empregados poderia incluir o centro de custo, título, assinatura de autorização, e outra informações úteis para o controle de acesso de granulação fina. Por outro lado, algumas organizações centralizam a administração e às informações de um usuário em um diretório baseado em LDAP como o "*Oracle Internet Directory"*. Podem ser armazenados atributos de contexto de aplicação no diretório e podem ser concedidos a um ou mais usuários. Estes podem ser recuperados automaticamente no login de um usuário, e usados na inicialização de um contexto de aplicação.

- Contexto de aplicação Inicializado Globalmente:

Esta característica fornece uma localização central para o armazenamento do contexto da aplicação do usuário, enquanto deixa a aplicação montar os contextos do usuário, durante a inicialização, formando a identidade do usuário. Em particular, apóiam rótulos do "*Oracle Label Security"* e privilégios. Isso, facilita a administração de contextos de grande número de usuários do banco de dados.

Para inicialização do contexto de aplicação global é necessário utilizar o LPAD, que guarda a lista de usuários e a qual aplicação esta associada. Caso não se tenha o LDAP pode ser usado o "*Oracle Internet Directory*" para autenticação e autorização dos usuário Oracle.

- Contexto de aplicação teve acesso globalmente:

O contexto de aplicação global, pode ser compartilhado entre sessões com relacionamento de confiança. Adicionando políticas de controle de acesso de granulação fina, para conduzir a utilização das aplicações (especialmente em produtos de camada do meio), podem usar atributos global para suportar aplicação segura.

Várias aplicações baseadas em Web, usam agrupar conexões para aumentar o nível da capacidade de conexão, através disso, suportar centenas ou milhares de usuário. Estas aplicações reusam conexões, ao invés de para cada usuário ter uma sessão diferente. Por exemplo, dois usuários A e B conectam-se a uma aplicação de camada do meio a qual estabelece uma sessão com o banco de dados usando esta em nome dos dois usuário. A aplicação é responsável para alterar o usuário na conexão, de tal forma, que em qualquer momento tanto o usuário A ou o B possam usar a sessão.

O VPD facilita o agrupamento de conexão, permitindo múltiplas conexões, tendo acesso a um ou mais contextos de aplicações globais, em vez de montar um contexto de aplicação para sessão de usuário. Os contextos de aplicações globais disponibilizam flexibilidades adicionais para aplicações baseadas em Web usando aumentar o desempenho do controle de acesso através do reuso, um contexto de aplicação comum entre várias sessões, no lugar de montar um contexto de aplicação para cada sessão.

Aplicações de usuários com autenticação por procuração, podem ser usadas com contexto de aplicações globais para adicionar flexibilidade e aumentar a execução na construção de aplicação de eBusiness. Por exemplo, imagine uma aplicação baseada em Web, que disponibiliza informações para empresários com três níveis de usuários: Ouro, Prata e Bronze, representando níveis diferentes de informações. Em lugar de cada usuário ter sua própria sessão com contextos de aplicações individuais, ao ativar, a aplicação poderia montar contextos de aplicações globais para Ouro, Prata o Bronze e poderia usar um identificador de cliente para indicar a sessão para o contexto correto e em ordem retornar os tipos de dados apropriados. A aplicação, precisa inicializar somente uma vez, os três contextos globais, para limitar o acesso aos dados.

4.4.3.5 Como dividir o controle de acesso de granulação fina para facilitar VPD

O controle de acesso de granulação fina permite a construção de aplicações que exijam políticas de segurança com um baixo nível de granulação. Por exemplo, pode-se usar isso, para restringir o acesso de um cliente, para que ele veja somente sua própria conta, ou seja, um médico para ver apenas os registros dos seus pacientes, ou o gerente ver só os registros dos empregados que trabalham com ele.

A habilidade para dividir a execução de políticas de segurança através do VPD facilita o desenvolvimento de aplicação. Por exemplo, supondo uma aplicação de entrada de pedidos e uma aplicação de inventário acessando a mesma tabela de pedidos. A aplicação de entrada de pedidos, limita o acesso baseado no número do cliente, enquanto que a aplicação de inventário limita o acesso baseado no número da peça. É muito útil poder dividir o controle de acesso de granulação fina de forma a aplicar diferentes políticas de segurança, de acordo com a aplicação que se está acessando os dados. Caso contrário, o desenvolvedor da aplicação de entrada de pedidos e respectivamente de inventário, teria que aplicar uma política comum para duas, podendo ser possível ou não. Com o VPD as aplicações podem ter diferentes políticas de segurança de acordo com as necessidades individuais de cada aplicação.

O Oracle9i possibilita o particionamento do banco de dados privado virtual através de grupos de políticas e um condutor de contexto de aplicação. Um condutor de contexto de aplicação, determina com segurança qual aplicação está acessando os dados, e grupos de políticas, facilitando a administração de políticas que são executadas nas aplicações. O SGBD Oracle também suporta padrões de grupo de políticas, os quais sempre são aplicadas aos dados.

4.4.3.6 Modelos de usuários e banco de dados privado virtual

Aplicações podem ter diferentes modelos de usuários. Existem vários modos nos quais as aplicações podem garantir o controle de acesso de granulação fina, independente de o usuário ser conhecido ou desconhecido para o banco de dados.

Nas aplicações onde o usuário é o mesmo do banco de dados, o uso do VPD é realmente simples, no momento que o usuário conecta-se ao banco, este pode montar contexto de aplicação para cada sessão, de forma que, cada sessão é montada sobre um identificador de usuário simplificando o controle de acesso de granulação fina, para diferentes usuários. Isto também é possível quando a autenticação é por procuração, visto que cada sessão JDBC-OCI é fraca "leve", esta ainda é uma sessão distinta no banco de dados e pode ter seu próprio contexto de aplicação. Isso, facilita quando a autenticação de procuração é feita pelo LPAD ou "*Oracle Internet Directory*", onde pode retornar papéis dos usuários, bem como, outros atributos para execução do VPD.

Para aplicações que utilizam um usuário único para conecta-se ao banco, em nome de todos os usuários, mesmo assim é possível fazer o controle de acesso de granulação fina. Mas, para isso, o desenvolvedor da aplicação pode criar um atributo de contexto que representa o usuário da aplicação (como UsuarioReal). A sessão é inicializada com o contexto do usuário único, mas pode ter atributos que variam dependendo quem é o usuário real. Este modelo trabalha melhor para aplicações com um número limitado de usuários, onde às exigências não variam muito para cada nova sessão.

4.4.4 Gerente de política de Oracle

Com o gerente de política Oracle melhora a administração das políticas de VDP, isso através de uma interface gráfica (GUI) de fácil uso o qual pode ser acessado pelo "*Oracle Enterprise Manager*". O desenvolvedor pode usar gerente de política da Oracle para aplicar políticas de segurança a objetos do esquema, como tabelas e visões, bem como, criar contextos de aplicações, desta forma, a estrutura VPD facilita muito desenvolver e administrar. O gerente de políticas Oracle também é uma ferramenta para "*Oracle Label Security*", um produto baseado em VPD que prove acesso a dados baseado em rótulos. Segurança de rótulos da Oracle é uma solução genérica para o problema de controle de acesso aos dados com granulação fina.

4.4.5 Controle de acesso baseado em rótulo

Um rótulo adiciona um sofisticado controle de acesso de regras, além dos fornecidos pelo controle de acesso de discriminatório. Além disso, o controle de acesso baseado em registros de dados pode identificar o rótulo do usuário e do registro. Com isto, garantindo um nível adicional de controle de acesso a um sistema (LEVINGE,2002b).

Controle de acesso baseado em rótulo permite que as organizações nomeiem rótulos de sensibilidade para registros de dados, este controlo de acesso ao banco de dados com estes rótulos, assegura uma sensibilidade apropriada para aqueles dados que estão marcados com o rótulo. O exemplo mais comum disto é talvez o sistema de classificação de segurança usado pelos Estados Unidos e outros governos. Neste modelo, as etiquetas são classificadas hierarquicamente como CONFIDENCIAL, SEGREDO, ou SUPERSEGREDO e, é nomeada com sensibilidade ao nível da informação no banco de dados. Além, das classes de segurança formal estão definidos e nomeados aos dados, como NATIVO ou CRITOGRAFADO. Ter acesso a dados etiquetados com um certo nível (como SEGREDO) a estes é restringido o acesso apenas aos usuários, aos quais foram concedidos níveis de acesso iguais ou mais altos. Ter acesso aos dados num depósito específico (como NATIVO CONFIDENCIAL) é restringido a esse quem têm acesso apropriado neste nível, assim como, permissão explícita de acesso para o compartimento em questão (LEVINGE,2002b).

Enquanto sistemas tipicamente eBusinesses não tem rótulo classificando dados, eles quase sempre exigem que tenham dados rotulados. Por exemplo, um eBusinesses pode diferenciar entre " informação confidencial da companhia " e " informação de público ". Mais adiante, poderá haver alguma informação confidencial da companhia que pode ser compartilhada com sócios, debaixo de um acordo confidencial de revelação, ou outro documento legal, enquanto outras informações só poderão ser acessíveis por certos grupos dentro da companhia (como finanças ou divisões de vendas). A habilidade para gerenciar rótulos de dados nativos é uma grande vantagem

para eBusinesses podendo prover a informação certa para às pessoas certas ao nível certo de segurança de acesso aos dados.

Um rótulo adiciona um sofisticado controle de acesso de regras, além dos fornecidos pelo controle de acesso de discriminatório. Além disso, o controle de acesso baseado em registros de dados pode identificar o rótulo do usuário e do registro. Com isto, garantindo um nível adicional de controle de acesso a um sistema.

O controle de acesso baseado em rótulo depende da política básica de DAC, essas políticas juntas ditam os critérios pelos quais o acesso ao objeto é permitido ou negado. Na maioria das aplicações, um número relativamente pequeno de tabelas requer um controle de acesso baseado em rótulo. A proteção provida pelo DAC basta para maioria da tabelas da aplicação (LEVINGE,2002b).

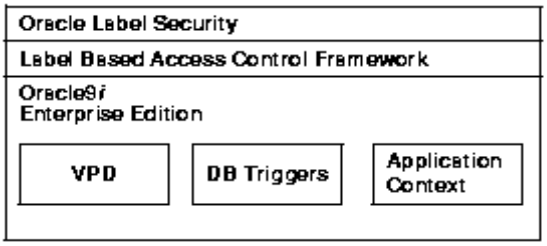
- Como controle de acesso Rótulo-baseado trabalha com DAC:

Eles trabalham integrados, para um usuário ter acesso a um registro de uma tabela, primeiro ele deve ter os privilégios do DAC para executar tal tarefa na referida tabela, só após isso, o usuário deve conhecer os critérios de segurança baseada em rótulo. A segurança do Oracle baseada em rótulo adere as definições de rótulos, de rótulos hierárquicos, e outras regras de políticas de segurança definidas pelos administradores nos bancos de dados locais.

4.4.5.1 Arquitetura da segurança baseada em rótulo

A segurança baseada em rótulo é construída sobre o VPD. Também usa a funcionalidade de contexto de aplicação.

Figura 11 - Oracle Label Security and Oracle9i Enterprise Edition

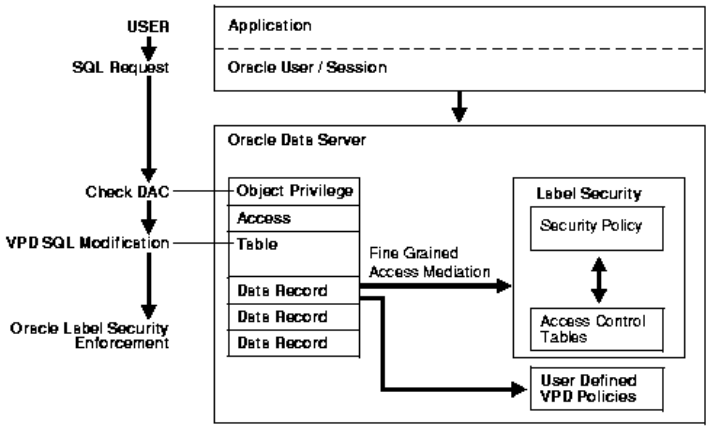


Fonte: Oracle Label Security Administrator's Guide, Release 2 (9.2)

A segurança baseada em rótulo da Oracle é um *framework,* que aumenta a possibilidade de implantar segurança ao nível de registros. Ela possibilita a fácil criação de políticas baseadas em rótulos, sem qualquer conhecimento de programação. Bem como, não existe a necessidade de se escrever uma linha se quer de código, em um único passo pode-se aplicar uma política de segurança a uma determinada tabela. Deste modo, segurança baseado em rótulo disponibiliza um modo direto e eficiente de implantar políticas de segurança de granulação fina que usam a tecnologia de etiquetar dados (LEVINGE,2002b).

Cada registro de dados tem um rótulo; e a segurança de rótulo é invocada para cada registro, isso determina, se o usuário pode ou não ter acesso ao registro.

Figura 12 - Arquitetura do Oracle Label Security



Fonte: Oracle Label Security Administrator's Guide, Release 2 (9.2)

4.4.5.2 Características da segurança baseada em rótulo da Oracle

A segurança baseada em rótulo prove um controle de segurança adicional ao nível de registros.

Por exemplo, digamos que um usuário possui um privilégio de selecionar dados de uma tabela de uma determinada aplicação. Quando o usuário executar uma requisição de "*select*", a segurança baseada em rótulo avalia cada registro selecionado e determina se o usuário irá poder ter acesso baseado nos privilégios e rótulos nomeados ao usuário pelo administrador de segurança. Da mesma forma, pode ser configurada segurança para tarefas de modificação, eliminação e inserção de registros em tabelas do banco de dados (LEVINGE,2002b).

Figura 13 - Segurança baseada em rótulo.

Título	Preco	Rótulo Registro
Chefe	600	NaoClassificado
Coordenado	400	NãoClassificado
Diretor	750	SuperSecreto
Gerente	600	Secreto
Coordenado	450	Secreto

Segurança baseada em rótulo habilita um conjunto de privilégios de autorização de acesso. O rótulo assegura que os dados possam ser protegidos independente de seu próprio valor.

Segurança baseada em rótulo fornece políticas flexíveis para controles exigentes de processamentos especiais.

Podem ser protegidas tabelas específicas da aplicação. Não é necessário aplicar políticas de segurança em todas tabelas da aplicação, apenas quando necessário. Por exemplo, tabela de CEP não é necessário.

Segurança baseada em rótulo permite o administrador de segurança adicionar funções rotuladora especial e determinados SQL na política.

Podem ser criadas várias políticas de segurança baseada em rótulos. Por exemplo, uma política de recursos humanos poderia co-existir com uma política de defesa no mesmo banco de dados. Cada uma das políticas pode ser configurada independentemente, e ter um único rótulo definido.

4.4.5.3 Características do Framework de politicas de rótulos

O *framework* adiciona ao SGBD objeto-relacional controles de acesso baseados em rótulos. O controle de acesso tem como os seguintes fatores:

- O rótulo associado a um registro de dados.
- O rótulo associado com uma sessão de usuário.
- Os privilégios de política associados a uma sessão de usuário.

As opções de políticas de execuções associadas a uma tabela.

Por exemplo, considere execução de um comando padrão DML (como selecionar) em um registro de dados. Ao avaliar este pedido de acesso de um usuário com o rótulo CONFIDENCIAL, para um registro de dados etiquetado como CONFIDENCIAL, a segurança baseada em rótulo determina que este acesso pode realizado.

Desta forma, dados com diferentes sensibilidades, pertencendo a diferentes companhias, podem ser armazenadas e administradas em um único sistema, preservando a segurança de dados por controles de acesso padrão. Do mesmo modo, que aplicações industriais de larga escala podem usar rótulos nos registros provendo um adicional controle de acesso e funcional de acordo com a necessidade (LEVINGE,2002b).

4.4.5.4 Rótulo de dados

Cada registro de uma tabela pode ser etiquetado com seu nível confidencial. O rótulo contém três componentes:

Um único nível ou posição de sensibilidade.

Um compartimento mais horizontal ou categorias.

Um grupo mais hierárquico.

O nível especifica a sensibilidade dos dados. Uma organização de governo típica poderia definir níveis, CONFIDENCIAL, SENSÍVEL, e SUPER_SENSIVEL. Já uma organização comercial poderia definir um único nível para dados de COMPANHIA_CONFIDENCIAL. O componente de compartimento não é hierárquico; são definidos compartimentos tipicamente para segregar dados, como dados relacionados a uma contínua exigência estratégica. Finalmente, são usados grupos para registrar propriedade e podem ser usadas hierarquias. Por exemplo, FINANÇAS e ENGENHARIA são grupos que podem ser definidos como filhos do grupo CEO, criando uma relação de propriedade (LEVINGE,2002b).

Os rótulos podem conter apenas nível de componente, um nível combinado com um jogo de compartimentos ou grupos, ou um nível com compartimentos e grupos.

- Rótulos de autorizações.

Podem ser concedidos rótulos de autorizações aos usuários, o que determina qual o tipo de acesso (leitura ou escrita) que eles têm nos registros que estão rotuladas.

- Política de privilégios.

Política de privilégios permitem que usuário ou unidade de programa armazenado possam contornar as políticas de controle de acesso baseado em rótulo. O administrador pode autorizar o usuário ou unidade de programa para executar ações específicas, como a habilidade de um usuário de assumir as autorizações de um usuário diferente.

Podem ser concedidos privilégios para unidades de programas, permitir o processamento, em lugar de o usuário, executar operações privilegiadas. Quando somente unidades de programas armazenados, e não os usuários individuais têm privilégios de segurança baseada em rótulos, seu sistema está mais seguro. Além disso,

tais unidades de programa encapsulam a política, o qual minimiza a quantia de código de aplicação que precisa ser revisado para segurança.

- Aplicação de opção de políticas.

Em aplicações baseadas em rótulo podem ser aplicadas diferentes opções para maior flexibilidade no controle das diferentes operações em DML executas pelos usuários. Para cada execução de operação de selecionar, inserir, modificar ou eliminar pode ser especificado um tipo particular de política de segurança para cada tabela. Deste modo, os controles de acesso baseados em rótulos podem ser personalizados para cada tabela.

4.5 CONCLUSÃO

Dentro deste capítulo, apresentou-se a forma especifica de como o SGBD Oracle de fazer o controle de privilégios, tanto de sistema como de objetos, apresentando principalmente os controle com a utilização de papeis, principalmente devido a este implementar o uso de papeis de banco de dados, globais, empreendimento e de aplicações seguras.

Os papeis de banco são definidos diretamente no banco em questão, já os globais são definidos nos serviços de diretórios com LDAP, os de empreendimentos são utilizados para empresas que compartilham informações entre diversos banco de dados. Os de aplicações seguras criam um mecanismo que da possibilidade de os privilégios serem concedidos em tempo de execução pelas aplicações, tendo estes a possibilidade de solicitarem senhas.

Foram apresentadas as possibilidades dos controles de acesso com a utilização de storage procedure onde aos usuários são apenas concedidos os privilégios de executarem estas e não ter acesso às tabelas diretamente, com isso garantindo que as modificações das tabelas sejam feitas apenas de acordo como for projetado nas storage procedures.

Segurança ao nível de registro é outra forma que o Oracle implemanta no processo de controle de acesso sendo que este dá a possibilidade de controles baseados em dados para isso o Oracle utiliza-se de um mecanismo de reescrita de consultas chamado de VPD

Outra forma de garantira a confidencialidade dos dados é a utilização de rótulos de segurança implementada pela Oracle, esta cria a possibilidade de restringir o acesso aos dados pela sua sensibilidade.

5 PESQUISAS REALIZADAS SOBRE CONTROLE DE ACESSO EM SGBD

5.1 INTRODUÇÃO

Controle de acesso em SGBD's não é um tema novo. Vários pesquisadores já escreveram sobre o assunto, mas entre os pesquisadores nacionais não foram encontradas publicações sobre o mesmo. Encontra-se pouca literatura traduzida na área de controle de acesso de banco de dados, apenas são abordadas algumas páginas em um ou dois capítulos das bibliografias sobre SGBD, onde são feitos apenas comentários básicos sobre o tema. O que foi encontrado sobre este assunto está relatado no segundo capítulo: Controle de acesso padrão do SGBD.

Encontramos vários artigos sobre o assunto, os mesmo não são muito recentes. A maioria dos artigos foram encontrados na: IEEE Computer, ACM Association for Computing Machinery, CiteSeer The NEC Research Institute Scientific Literature Digital Library e, alguns em outros endereços de menos relevância.

Dentre o assunto pesquisado, foram encontrados duas linhas, a dos controles de acesso discriminatório e a dos controles de acesso obrigatório (MAC).

No controle de acesso discriminatório as pesquisas se concentram no controle baseado em papéis. Dentro deste, estão sendo pesquisadas várias extensões para aumentar sua facilidade e flexibilidade no processo de concessão de privilégios aos

usuários. Conforme artigos apresentados por (SANDHU,1994), (SANDHU,1996), (SANDHU,1997), (SIMON,1997), (BERTIN0,1999), (OSBORN,2000).

> Conforme (SANDHU,1994) “O conceito de controle de acesso baseado em (RBAC) começou com o multi usuário e multi aplicação em sistemas on-line que abriram caminho nos anos setenta. A conceito principal de RBAC é que permissões são associadas com papéis, e são nomeados aos usuários estes papéis. Isto facilita e simplifica a administração de permissões. São criados papéis para as várias funções de trabalho em uma organização e são nomeados os usuários aos papéis baseado nas responsabilidades e qualificação deles”.

No processo de controle de acesso obrigatório as pesquisa estão voltadas para o controle baseado em rótulos e mecanismos de segurança em multi nível (MLS).

5.2 MODELOS DE CONTROLE DE ACESSO DISCRIMINATÓRIOS PESQUISADOS

Abaixo são apresentadas pesquisas realizadas em relação aos controles de acesso discriminatório em banco de dados relacionais.

5.2.1 Limitações de controles discriminatórios

Os controles de acesso padrões de SQL são considerados discriminatórios, porque a concessão dos privilégios é feita diretamente aos usuários.

Está provado que os controles de acesso discriminatórios tradicionais, são inadequado para as necessidades de segurança de muitas organizações. Da mesma forma, também são percebidos que os controles de acesso obrigatórios baseado em rótulos de segurança são impróprios para muitas situações. Nos últimos anos a noção de

controle de acesso baseado em papéis (RBAC) emergiu como um candidato para preencher a lacuna entre DAC tradicional e MAC, (OSBORN,2000).

Mesmo que for cuidadosamente controlado a concessão do acesso a uma tabela, para um usuário com acesso SELECIONAR, este tem a possibilidade de criar uma cópia da tabela, e assim poder burlar todos estes controles.

Além disso, mesmo que os usuários forem de confiança é possível que isso ocorra, caso o usuário use um programa infetado por um Cavalo de Tróia que acaba fazendo tal processo (SANDHU,1994).

Outra fraqueza de DAC em SQL é que este não facilita a administração de direitos de acesso. Para cada usuário deve ser concedido todo privilégio que eles precisam para realizar explicitamente as suas tarefas. Freqüentemente grupos de usuários têm a necessidade de privilégios semelhantes ou idênticos.

Para este caso, controle acesso baseado em papéis (RBAC) permite a criação de papéis para estes grupos de usuário. A estes papéis são concedidos privilégios explícitos. Então são associados os usuários aos papéis apropriados, dos quais eles herdam estes privilégios. Isso, responde duas perguntas, que privilégios deveriam receber um papel e, qual usuário deveria ser autorizado para cada papel. Com o uso de RBAC fica mais fácil designar um usuário a um papel novo ou colocá-lo em outro, ou alterar os privilégios de uma papel existente (SANDHU,1994), (SANDHU,1996), (BERTIN0,1999), (OSBORN,2000).

5.2.2 Uma política baseada em papéis para modelo de objetos

De acordo com LUPU, 1997 os papéis da empresa definem os deveres e responsabilidades dos indivíduos que são nomeados a eles. O artigo apresentado por ele, introduz um “framework” para a administração de grandes sistemas distribuídos, que fazem uso dos conceitos desenvolvida em teoria de papel. O conceito de papel para ele é o agrupamento e a especificação de políticas de administração de direitos e deveres que correspondem aos do papel de um indivíduo na empresa. Desta forma, indivíduos

podem ser nomeados e retirados de papéis, com isso, alterando seus direitos e deveres de modificar os papéis, deixando o processo mais flexível de definição de políticas. Foi estendido este conceito de papel para incluir relacionamentos como meios de especificar interações exigidas, deveres e direitos entre papéis relacionados. Empresas podem conter grandes números de papéis semelhantes com vários relacionamentos entre eles, assim, existe a necessidade de reuso de especificações. É permitido várias instâncias e herança de classes de papéis e relacionamentos para incremento da extensão da estrutura da empresa com mínimo de esforço na especificação. Nós examinamos consistência, também brevemente e examinamos assuntos relacionados a este "framework" de papel.

5.2.3 Mecanismo de autorização flexível para SGBD relacional

(BERTINO,1996a), apresenta um modelo de autorização para autorizações com tempo não selado. Este modelo apóia autorizações positivas e negativas e permite administração descentralizada. Foi apresentado com autorizações positivas e negativas coexistir. Discutiu-se o feito das operações administrativas na declaração de autorizações.

Em (BERTIN0,1999) é apresentado outro modelo de autorização, que pode ser usado para expressar várias políticas de controle de acesso discriminatório para SGBD relacional. O modelo permite autorizações positivas e negativas e apóia exceções ao mesmo tempo. O modelo é bastante flexível de acordo com o que os usuários podem especificar. Para cada autorização concedida, pode ser aplicado autorizações fortes ou permitir exceções. O mesmo prevê administração de autorização por grupos, podendo usar exceções a qualquer nível da hierarquia de grupo, com isso, disponibilizando suspensão temporária de autorizações. O modelo apóia a administração descentralizada de autorizações com o uso de propriedade. Também podem ser restringidos privilégios administrativos, de forma que, os donos retenham o controle sobre suas tabelas.

No modelo, podem ser especificadas autorizações para usuários, como também, para grupos de usuários. Não há distinção de forma de concessão de privilégios só para usuários ou grupos, mas pode haver junções de ambas as possibilidades. A forma de

concessão é parecida com a semântica de grupos de usuário, diferente de papéis (que são dinâmicos, onde o usuário pode fazer parte de um papel e, em outra situação pode fazer parte de outro papel conforme melhor convier), os grupos são estáticos, para que o usuário do grupo sempre faça parte dos membros do grupo. Embora, possam ser especificadas autorizações para usuários e grupos, o controle de acesso sempre opera como se os privilégios fossem concedidos diretamente ao usuário.

A introdução de ambas as autorizações fortes, na qual não permitem exceções e autorizações fracas que permitem exceções, aumentou a expressividade do modelo. Neste, não existe distinção entre autorizações fortes e fracas. Desta forma, todas as autorizações se comportam como forte ou como fraca. No primeiro caso, não seria possível apoiar exceções nas autorizações. No segundo caso, não seria possível obrigar com certeza as autorizações.

Podem ser resumidos os princípios básicos do modelo como segue:

Autorização pode ser negativa, positiva, forte ou fraca.

Autorizações fortes sempre devem ser obedecidas. Conseqüentemente, um assunto não pode conter autorização positiva e negativa ao mesmo tempo.

Se um assunto receber uma autorização negativa e uma positiva para o mesmo acesso, de tal forma, que uma é forte e outra é fraca, a autorização forte anula a autorização fraca.

Se um assunto recebe autorização negativa e uma autorização positiva fraca de acesso, a autorização pode anular a outra dependendo se este for ou não membro de um grupo e suas autorizações.

A um assunto, só é permitido o acesso, se ele tiver uma autorização forte positiva ou uma autorização fraca positiva por não se anularem.

5.2.3.1 Definição formal do mecanismo de autorização flexível

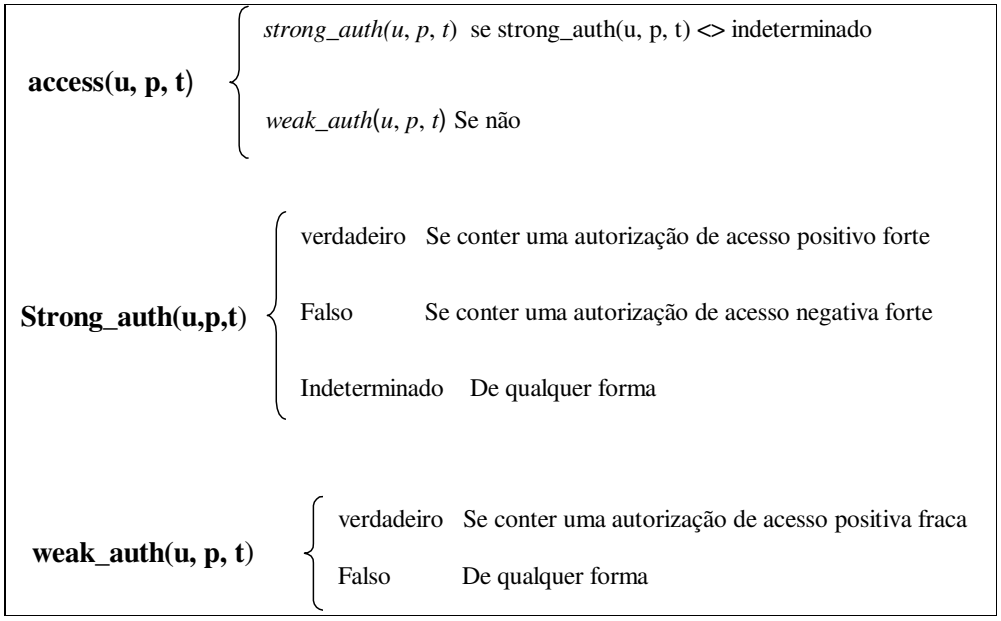
Na abordagem apresentada, quando for concedido um privilégio de acesso forte a um usuário e para o mesmo for concedido um privilégio fraco, o forte prevalece sobre o fraco. Da mesma forma, se for concedido um privilégio forte de acesso e um forte de negação o acesso não será aceito. No caso onde existir duas concessões, uma positiva e uma negativa, mas, ambas fracas, neste caso não é obvio o que deve ser feito. No modelo apresentado por (BERTIN0,1999), a decisão de qual das autorizações deve ser feita (se qualquer) está baseada no conceito da autorização mais específica. Desta forma autorizações feitas diretamente a usuários são mais específicas do que autorizações feitas para grupos. Isso, ocorre desta forma, por que um determinado privilégio pode ser negado por um caminho de grupo e por outro este pode ser concedido, neste caso, o que for mais específico prevalecerá.

5.2.3.2 Controle de acesso do mecanismo de autorização flexível

Conforme apresentado por (BERTIN0,1999), o controle de acesso está baseado no conceito de autorização válida. A autorização de privilégio em um objeto só será considerada válida se existir uma autorização feita para este e não existir nenhuma autorização que conflitar com a mesma para o mesmo objeto. O pedido de acesso será concedido se, e somente se, existir uma autorização positiva válida para tal.

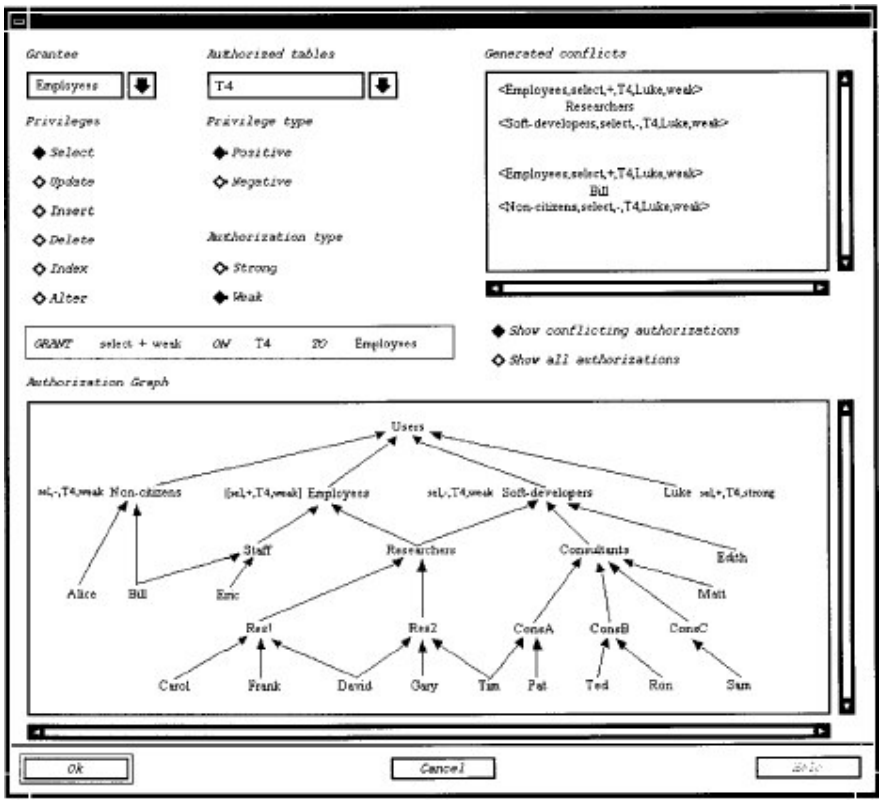
Foi apresentado controle de acesso feito por uma função “acess()”, na qual, para verificar necessita dos seguintes parâmetros, “U,P,T”, onde U é o usuário que está solicitando o privilégio P, para executar na tabela T, será concedido o direito em caso verdadeiro e caso, contrário será falso. A função foi definida conforme segue:

Figura 14 - Fórmula de acesso do Mecanismo de autorização flexível



Para comprovar a abordagem (BERTIN0,1999) desenvolveu um protótipo do sistema de autorização. Na implementação foi simulado o catálogo de uma SGBD comercial, no qual foi desenvolvido o modelo proposto. Também foram desenvolvidas ferramentas de administração dos privilégios. Segue abaixo imagem da ferramenta de administração desenvolvida.

Figura 15 - Imagem da ferramenta desenvolvida Mecanismo de autorização flexível.



Fonte: Artigo - A Flexible Authorization Mechanism for Relational Data Management Systems. De Bertino 1999.

5.2.4 Extensão das operações de concessão e revogação em SQL, para limitar e reativar privilégios

(ROSENTHAL,2000) desenvolveu uma proposta de extensão do processo de concessão e revogação de privilégios em SQL, com o objetivo de poder impor limites aos concessores permitindo, reativar privilégios em períodos determinados.

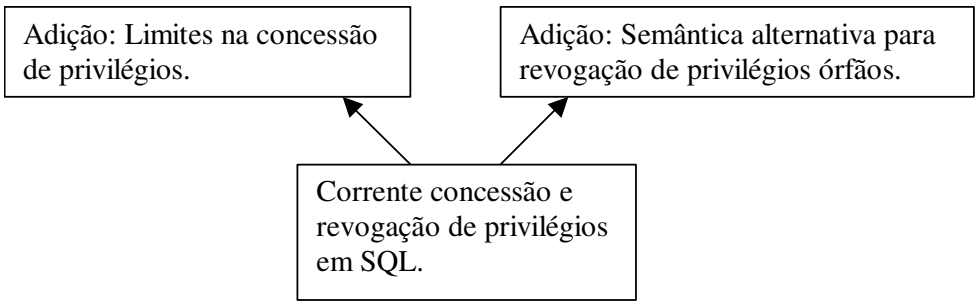
Para isso, ele propôs duas extensões ao modelo de segurança nos comandos de concessão e revogação (grant/revoke) em SQL. Segundo (ROSENTHAL,2000) “Em SQL, as concessões são incondicionais”, desta forma, o concessor simplesmente deve ter confiança em que está recebendo. Em uma das extensões é, permitido ao concessor

impor limitações na forma como o privilégio será usado. Na segunda, propôs-se novos meios para, seletivamente, reativar permissões que foram revogadas.

De acordo com (ROSENTHAL,2000) "O modelo de segurança em SQL teve poucas extensões nos últimos 20 anos, com exceção da recente adição de controles de acesso baseado em papéis". Em seu texto, trabalhou-se os seguintes problemas:

- A necessidade de limitações de privilégios: Quando é delegado o poder de conceder privilégios a outros usuários principalmente em sistemas distribuídos, a estes deveria poder prover orientações e limites na forma como o poder possa ser exercido. Ao mesmo tempo, o mecanismo de limitação deveria respeitar linhas de jurisdição, e introduz no mínimo novas complexidades.
- A necessidade de flexibilidade na revogação de privilégio e opcional reativação: Há várias semânticas possíveis, de como órfãos são definidos e tratados.
- A necessidade de incluir visões como parte do modelo: estas deveriam se comportar da mesma forma que as tabelas básicas.

Figura 16 - Extensões propostas por Rosenthal.



5.2.4.1 Concessão de conceder privilégios com limitações

Foram seguidos três princípios no desenvolvimento desta extensão sendo eles:

- O sistema deveria unificar, em um meio flexível para limitar privilégios, em lugar de vários mecanismos sobrepostos.
- A habilidade para concessão deveria respeitar as cadeias naturais de autoridade.
- Determinação de limitações deveriam ser independentes do tratamento de reativação, tanto para semântica como para implementação.

O mecanismo é bastante simples, e está baseado em passar alguns parâmetros chamados de predicados para o processo de controle de acesso, bem como, para o controle de concessão de acesso. Este predicado é uma função booleana sem efeitos adicionais. As entradas desta função podem ser o concessor, beneficiado, tempo para concessão, etc. Não foi proposto uma sintaxe particular para o predicados, no SGBD's em SQL existe um número considerável de variáveis, que podem ser usadas como predicados e que trarão ótimos resultados.

Exemplo. Um predicado de tempo poderia avaliar se é usado durante horas de trabalho normais ou se o usuário trabalha noturna: ($TIME entre 8am e 6pm) ou ($USER em NightShift).

Poderia ter um predicado que limita futuras concessões obrigando estas a que pode ser repassado o privilégio, usando, por exemplo: $GRANTEE.

Exemplo: Um predicado poderia verificar, se falsa, toda vez que há uma concessão para usuários em particular: **not ($GRANTOR = 'Boris' and $GRANTEE = 'Natasha')**

Não foram levantados todos os predicados possíveis e necessários, apenas deu-se uma idéia do que é possível com o uso destes.

5.2.4.2 Semântica dos comandos de concessões

Como em um comando normal de concessão estão embutidos dois comandos, o comando básico, que concede o privilégio para executar uma tarefa específica e o

comando que concede o poder de concessão deste privilégio para outros usuários. (ROSENTHAL,2001) representou os predicados da seguinte forma, para limites de privilégios básicos “bpred” e para limites de privilégios de concessão “gpred”.

Segue alguns exemplos de como poderiam ser os comandos de concessão de privilégios:

A seguinte concessão, permite que João leia da tabela de SalarioInfo durante horário de trabalho:

```
grant select on SalarioInfo to João bpred ($TIME between 8am and 6pm)
```

A seguinte concessão permite João executar comandos de concessão em SalarioInfo, mas, só para contadores. Além disso, qualquer privilégio básico concedido deve ser executado durante horário de trabalho:

```
grant onward select on SalarioInfo to joe
gpred ($GRANTEE in Contadores)
bpred ($TIME between 8am and 6pm)
```

A semântica do SQL para repassar concessão de conceder o predicado pode ser obtido usando a sintaxe “with grant option”.

```
grant select on SalarioInfo to joe
gpred ($GRANTEE in Contadores) with grant option
bpred ($TIME between 8am and 6pm)
```

5.2.4.3 Revogação de privilégios com predicados de limitações

No processo de concessão de privilégios são formados grafos de concessão demarcando a trajetória das concessões, quando um nó deste grafo é interrompido os nós dependentes destes são revogados. Mas, como nos mecanismos atuais, quando há uma interrupção, esta ocorre com a revogação de algum nó, gerando um processo em cascata.

Na extensão proposta, este grafo pode ser quebrado por vários motivos, o normal é por um comando de revogação de privilégio, mas pode ser quando um predicado

específico ficar inválido, tornando toda cadeia de privilégios inválida. Desta forma, forçando o processo ser reavaliado, toda a vez que algum predicado ficar inválido.

Foram propostas algumas alternativas semânticas para o processo, mas que merecem um estudo maior.

Nunca reavaliar: Com isso, usando somente predicados de tempo como "gpred". Embora, isto exclui alguns predicados de limitação úteis, ainda é mais melhor que os atuais sistemas. Para imediata implementação, parece o mais desejável.

Cada concessão deve especificar quando seus predicados devem ser reavaliados: Esta aproximação tem flexibilidade considerável, mas requer tempo extra de qualificados administradores. Mesmo assim, pode se cometer enganos na definição de reavaliação.

Reavaliar continuamente: Todos os predicados são (conceitualmente) reavaliados depois de cada mudança de banco de dados. Se a cadeia de autorização de uma concessão ficou falsa, aquela concessão é revocada. Também são revocadas concessões adicionais, pelas quais esta concessão é responsável. Esta definição tem semântica simples, mas é impraticável de ser implementada.

Definir certos eventos importantes para reavalização: Só será reavaliado quando estes eventos ocorrerem.

(ROSENTHAL,2000) apresentou uma abordagem interessante ficando para serem estudadas alguns itens da mesma, bem como, o desenvolvimento dos algoritmos que irão dar sustentação ao processo.

5.2.5 Políticas de segurança flexíveis em SQL

(BARKER,2001) mostra como uma variedade grande de políticas de controles de acesso, baseados em papéis, pode ser formalmente especificada, utilizando-se de um subconjunto de cláusulas de lógica formal. Apresentado então como estas

especificações formais podem ser traduzidas automaticamente em um subconjunto pequeno de SQL, podendo ser usados para proteger bancos de dados relacionais de pedidos de usuários autenticados de leituras não autorizadas. Foi demonstrado o poder do controle de acesso, mostrando, assim, como uma variedade de políticas de controle de acesso pode ser representada em SQL.

Hoje, muitas organizações estão usando políticas de controle de acesso baseado em papéis. As organizações criam papéis, de acordo com o perfil e qualificações dos usuários associa estes aos papéis, a este ao concedidos privilégios de execução de funções em objetos. Desta forma, são nomeadas permissões em objetos para papéis, em lugar de diretamente para usuários, sendo o caso com políticas tradicionais, RBAC ajuda a simplificar as especificações de segurança de banco de dados e a administração.

(BARKER,2001) “Ainda produtos de SGBD e o padrão de SQL3 incluem um pouco suporte para papéis, existindo muitas políticas importantes que não podem ser expressas por estes produtos”.

5.2.5.1 Definição do modelo de Barker

Para entender este processo deve-se ter noção de como funciona o modelo RBAC que é, fornecer permissões especificas para execução de operações em objetos do banco de dados concedidos a papéis, e usuários associados a estes papéis. Esta, é apresentado no item “Papéis” destas, bem como por vários pesquisadores, entre eles (SANDHU,1993), (SANDHU,1994), (WINSLETT,1994), (SANDHU,1996), (SANDHU,1997), (SIMON,1997).

No modelo de (BARKER,2001), a representação da associação do usuário a um papel é feita da seguinte forma URA(U,R) e, a concessão de privilégios em objetos do banco para os papéis é assim representada RPA(R,P,O).

Onde nome URA é o apelido para o processo de associação do usuário ao papel (user-role assignment), na teoria de RBAC, a URA é usada para representar a associação de usuário U com o papel R. Da mesma forma, que RPA(R,P,O) é para

conceder privilégios em objetos do banco de dados para papéis (role-permission assignment), onde RPA é usada pra especificar a qual papel R é concedido que privilégio P em que objeto O do banco de dados.

Neste modelo pode existir o processo de hierarquia de papéis, o qual é representado pelo SENIOR-TO(R1,R2) que é a associação de um papel a outro papel, onde o papel R1 é associado ao papel R2.

Para verifica se o usuário tem um determinado direito, o modelo utiliza uma função PERMITTED(U,P,O) a qual verifica se ao usuário U foi concedido o privilégio P no objeto O do banco de dados. Para representar uma política aberta foi definido D-RPA(R,R,O), onde ao papel R é negado o privilégio P no objeto O do banco de dados. E, para auxiliar o processo de verificação de direitos foi necessário criar a função DENIED(U,P,O).

Desta forma, é possível de representar várias políticas de segurança abertas ou fechas com este método. O fato é que, um número pequeno de cláusulas de aplicação não consegue expressar estas necessidades específicas de políticas de segurança baseada em papéis (RBAC), o processo de manutenção e controle de acesso com o controle baseado em papéis é bastante simples. O interessante deste método é que o mesmo utiliza um número pequeno de comandos SQL para representear todo o processo. (BARKER,2001).

5.2.5.2 Modelo de Barker em SQL

Para entender o modelo em SQL será apresentada parte deste processo. Como objetivo de comprovar o seu método (BARKER,2001) criou tabelas para guardar os privilégios concedidos. Foram criadas as seguintes tabelas: URA, a qual armazenas a associação dos usuários aos papéis; A tabela RPA, a qual armazena os privilégios de execução de operações em objetos do banco de dados concedidos aos papéis; E a tabela SENIOR-TO, onde são armazenadas as associações de papéis em relação a outros papéis. Para juntar e poder controlar as permissões foi necessário criar a visão PERMITTED conforme segue;

```
CREATE VIEW permitted(U,P,O) AS
SELECT ura.U, rpa.P, rpa.O
FROM ura, rpa, senior-to
WHERE ura.R =  senior-to.senior AND senior-to.junior = rpa.R;
```

O mecanismo todo é muito simples o que é feito é apenas incluído aos comandos SQL feitos a base de dados, a verificação se este é válido para ser executado conforme veremos nos exemplos:

- Comandos de Selecionar uma tabela:

```
SELECT *
FROM t1
WHERE EXISTS(SELECT
FROM permitted
WHERE permitted.U = 'u' AND permitted.P = 'read'
AND permitted.O = 't1');
```

Comandos de Selecionar mais tabelas:

```
SELECT JNo
FROM SPJQ,S,P,J
WHERE J.JNo= SPJ.JNo AND SPJQ.PNo= P.PNo AND
P.Colour= 'red' AND SPJQ.SNo= S.SNo AND S.City= 'London' AND
EXISTS(SELECT
FROM permitted
WHERE permitted.U= 'Sue' AND permitted.P= 'read' AND permitted.O= 'SPJQ') AND
EXISTS(SELECT
FROM permitted
WHERE permitted.U= 'Sue' AND permitted.P= 'read' AND permitted.O= 'S') AND
EXISTS(SELECT
FROM permitted
WHERE permitted.U= 'Sue' AND permitted.P= 'read' AND permitted.O= 'P') AND
EXISTS(SELECT
FROM permitted
WHERE permitted.U= 'Sue' AND permitted.P= 'read' AND permitted.O= 'J');
```

Comandos de Updade:

```
UPDATE P
SET Color= 'yellow'
WHERE Color = 'red' AND
EXISTS(SELECT
FROM permitted
WHERE permitted.O = 'P' AND permitted.P = 'update' AND permitted.U = 'Sue');
```

RBAC é de fácil utilização e está presente em algum SGBDs, mas não conseguem representar todas as exigências de segurança que determinadas aplicações necessitam. Já o modelo de Barker tem a possibilidade de fazer o controle de acessos híbridos, pois tem a capacidade de controles de concessões positivas e negativas no mesmo mecanismo, além destes, podem ser controlados privilégios temporais, aqueles privilégios que um determinado usuário tem, mas em um período específico do dia,

juntamente com estes itens podem ser representados alguns mecanismos de impedir certas situações (constraints).

> A conclusão de (BARKER,2001) é que "O grande desafio para melhorar a formulação de políticas de controle de acesso é prover praticidades utilizáveis, construídas em alto nível e possuindo uma semântica clara. Nós mostramos como isso pode ser feito. A abordagem que descrevemos permite um DBA especificar uma variedade grande de políticas de controle de acesso usando elementos de RBAC mapeadas em lógica estratificada. Além disso, nós demonstramos isso pode ser construído e implementado usando um subconjunto pequeno de SQL3."

5.2.6 Configurando o controle de acesso baseado em papéis para políticas mandatárias e discriminatórias

No artigo de (OSBORN,2000), ele apresenta como pode ser configurado e utilizado o controle de acesso baseado em papéis para ter os mesmos efeitos que a abordagem LBAC. Dentro de seu trabalho ele apresenta um comparativo entre as duas abordagens, apresentando resultados das comprovações feitas.

O trabalho dele buscou mostrar a possibilidade, de com uma das abordagens pode ser representada a outra. Dentro do processo comprovou-se a importância da hierarquia de papéis no processo de controle de acesso como para simulação de LBAC. Mostrou-se também, que a administração da hierarquia de papéis é essencial pra execução de políticas de DAC.

(OSBORN,2000) conclui que "esta simulação de LBAC apenas, assume um papel administrativo, considerando que a simulação de DAC requer um grande número de papéis administrativos que são criados e destruídos dinamicamente".

5.3 MODELOS DE CONTROLE DE ACESSO OBRIGATÓRIOS PESQUISADOS

Neste item serão apresentadas pesquisas realizadas em relação ao controle de acesso obrigatório, este controle está presente no SGBD seguros ou com multi nível de segurança (MLS). Antes teremos que apresentar alguns conceitos.

5.3.1.1 Controle de acesso obrigatório

Controle de acesso obrigatório está baseado em rótulos de segurança associados a cada artigo de dados de cada usuário. Um rótulo, em um artigo de dados, é chamado de classificação de segurança, enquanto, um rótulo, em um usuário, é chamado uma liberação de segurança. Em um sistema de computador todo programa corrido por um usuário herda a liberação de segurança do usuário (SANDHU,1993), (SANDHU,1994), (WINSLETT,1994).

Segurança por rótulos forma em geral uma estrutura de grade. Para ficar mais fácil de exemplificar usaremos apenas dois rótulos S para segredo, U para não classificado. É proibida uma informação S ir para artigos dados U.

Existem duas regras de controles de acesso obrigatórias para alcançar este objetivo.

- Propriedade de Segurança Simples: Um U-usuário não pode ler S-dados.
- Propriedade Estrela: Um S-usuário não pode escrever U-dados.

Há alguns pontos importantes, que deveriam ser entendidos claramente neste contexto. Primeiramente, temos que entender que a regra assume que um usuário com liberação secreta, e poderá efetuar o login como S-usuário ou como U-usuário. Caso contrário a propriedade estrela, impedirá que um S-usuário escreva dados públicos ou U-dados. A segunda regra previne simplesmente que dados não possam ser lidos ou escritos nos dois sentidos.

Controles de acesso obrigatórios em bancos de dados de relacional geralmente obrigam uma propriedade de estrela mais forte dada abaixo.

Propriedade de Estrela Forte: Um S-usuário não pode escrever U-dados e um U-usuário não podem escrever S-dados.

A propriedade de estrela forte, limita cada usuário a escrever no seu próprio nível. Isto é considerado devido à integridade. A propriedade de estrela (fraca) permite para um U-usuário escrever S-dados. Esta propriedade dá a possibilidade de um U-usuário destruir os S-dados.

Rótulos de segurança podem ser associados aos dados no SGBD relacional em diferentes níveis de granulação. Nomear rótulos para tabelas inteiras pode ser fácil, mas geralmente são inconvenientes. Por exemplo, se a coluna salário de uma tabela for secreta, e as demais não, seremos forçados a colocar esta em outra tabela. Rótulos nomeados para toda coluna da mesma forma torna-se inconveniente.

A fina granulação de rotulagem é no nível de atributo individual de cada registro ou nível de elemento rotulado. Isto expressa a considerável flexibilidade. A maioria dos produtos emergentes nesta área oferece no nível de um registro. Embora, assim, não é tão flexível como ao nível de elemento rotulado, esta abordagem é mais conveniente que tabelas ou nível de colunas rotuladas (SANDHU,1994), (WINSLETT,1994).

5.3.1.2 Arquiteturas multi nível de banco de dados

Um sistema de multi nível é no qual usuários e dados com diferentes rótulos de segurança coexistem. Sistemas de Multi nível (os ditos) confiados, são aqueles que podem manter dados com diferentes rótulos separados, e assegura que são obrigadas propriedades de segurança estrela simples e (forte). Durante os últimos vinte anos, pesquisas consideráveis foram dedicadas no desenvolvimento de construção de bancos de dados de multi nível. Três arquiteturas viáveis emergiram como segue (SANDHU,1993), (SANDHU,1994).

- Arquitetura de dados integrada;
- Arquitetura de dados fragmentada;
- Arquitetura de dados replicados.

Os produtos de banco de dados relacionais que estão surgindo inicialmente nesta área, são arquiteturas basicamente de dados integradas. Esta abordagem requer modificações consideráveis nos SGBD relacionais existentes.

As arquiteturas fragmentadas e replicadas foram demonstradas em projetos de laboratórios por vários pesquisadores. Estas arquiteturas oferecem maiores garantias de segurança que a arquitetura integrada de dados.

5.3.1.3 Linguagem de consulta formal para SGBD relacionais seguros

(WINSLETT,1994), em sua pesquisa, apresenta uma linguagem formal para auxiliar o processo de busca de dados de forma segura, evitando os conflitos gerados pela abordagem de segurança em multi-nível. Sua pesquisa não aprofunda nenhuma forma de controle de acesso nesta abordagem, apenas auxilia o processo de melhoria na semântica da linguagem. Onde definiu uma álgebra relacional modal satisfatória para uso com semântica, mostrando como qualquer linguagem de consulta formal se transforma em uma consulta bem definida em MLS.

5.3.2 Fragmentação de dados para o controle de acesso

(DIDRIKSEN,1997) apresentou um controle de acesso aos dados via fragmentação dos mesmos. Para isso, o banco de dados é dividido em partes, conforme a necessidade de controle de acesso, a estes fragmentos são associados as obrigações (constraint) na forma de regras declarativas de acesso. O controle de acesso é baseado em papéis, onde cada usuário tem um papel a desempenhar dentro da empresa, com deveres e obrigações pertinentes ao cargo exercido.

Para o controle funcionar, foram desenvolvidos gatilhos de banco de dados que verificam os predicados associados a cada fragmento do banco de dados com os do papel de cada usuário. Desta forma, controlando que uma terminada operação poderia ser executada por determinado usuário no referido fragmento de dados, efetuando, desta forma, o controle de acesso.

Depois de analisados em um banco de dados, constatou-se que algumas tabelas deveriam sofrer a fragmentação dos dados para o controle de acesso. Baseado nestes tipos de fragmentação, foram gerados gatilhos que rejeitam transações, que tentam modificar dados fora do fragmento permitido.

O funcionamento é bastante simples, onde foram criadas algumas tabelas de controle das hierarquias de acessos, nas quais os eventos pegam informações para fazer os controles de acesso aos fragmentos de dados, de acordo com cada papel do usuário.

Este trabalho não está baseado em rótulos de dados e usuário, mas seu funcionamento é muito semelhante, mostrando que existem várias formas de se fazer o controle de acesso.

5.4 CONCLUSÃO

Dentro dos modelos de controle de acesso, podemos ter os controles discriminatórios e os controles obrigatórios. Ambos tem limitações para isso os desenvolvedores de aplicações juntamente com os responsáveis por segurança devem conhece-las, com isso criar mecanismos e políticas de segurança de acordo com cada tipo.

Muitas pesquisas estão sendo feitas para eliminar tais limitações, como com o uso de papéis, mecanismos que flexibilizam o processo de autorização dos SGBDs relacionais, criação de extensões para SQL, bem como criação de políticas flexíveis que garantam boa parte das exigências das aplicações. Estão sendo feitos estudos em SGBDs seguros onde são empregadas arquiteturas de multi nível de segurança, fragmentação de dados como processo de segurança.

Pode ser constatado que os dois modelos de controle de segurança sozinho não conseguem garantir todo o controle exigido pelas políticas de segurança, mas que quando aplicados em conjunto isso melhor em muito.

6 ANÁLISE DE ADERÊNCIA DAS POLÍTICAS DE SEGURANÇA AOS CONTROLES DE ACESSO EM SGBD

6.1 INTRODUÇÃO

Os controles de acesso dos SGBD's relacional na prática deveriam garantir as políticas de seguranças criadas pelas empresas. Estes controles estão aptos a controlar qualquer tipo política de segurança. Estes podendo não necessitar de controles adicionais. Os mesmos são de fácil administração.

De quem é a responsabilidade de desenvolver e controlar as políticas de segurança de dados em uma empresa? É do administrador da base de dados ou DBA. Esta afirmação é um tanto forte e difícil de ser comprovada, pois geralmente às informações que são armazenadas em um banco de dados, foram inúmeras pessoas de diferentes áreas ou até empresa que as armazenaram. Olhando pela ótica de que o responsável por um determinado ativo é o seu dono, poderíamos dizer que a responsabilidade pela informação é o dono da mesma.

Os SGBD's, quando em suas estratégias pré-estabelecidas, fazem com que o dono dos depósitos de informação (tabelas), é quem tem o direito de conceder privilégios sobre às informações nelas contidas, com isso, determinando que o responsável pela informação é o seu dono.

Mas, na prática a responsabilidade de desenvolver, criar e manter as políticas de segurança das empresas, bem como, criar e manter os mecanismos e meios necessários para que estas políticas sejam cumpridas conforme as mesmas determinam, é do administrador da base de dados.

No segundo capítulo, no item: determinação do problema, foram levantadas duas perguntas:

Como atender uma determinada política de segurança? Onde nesta política: um determinado usuário deve ter privilégios sobre um conjunto de registros de uma tabela específica quando estiver utilizando uma aplicação A, e um outro conjunto de registros da mesma tabela quando estiver acessando a aplicação B e, quando estiver acessando por qualquer outra aplicação, não deve ter acesso a referida tabela.

Os SGBD estão preparados para centralizar o controle de acesso garantindo as políticas de segurança das empresas sem a necessidade de mecanismos adicionais?

6.2 POLÍTICAS DE SEGURANÇA

Conjunto de normas e diretrizes destinadas a garantir a proteção dos ativos das empresas (ORANGE BOOK,1985), (CARUSO,1999), (MOREIRA,2001). Em resumo, devem ser definidas exigências de proteção em termos de ameaças percebidas, risco, e metas de controle de segurança de dados de acordo com uma empresa.

Uma política de segurança deve refletir as exigências de proteção, controle de utilização, processamento e manipulação das informações. A mesma tem como objetivo definir as expectativas da empresa, quanto ao uso dos seus recursos, estabelecendo procedimentos com o intuito de prevenir e responder a incidentes relativos a segurança.

Quando existem políticas de segurança bem definidas e implementadas, e corretamente seguidas temos como conseqüência os seguintes aspectos:

- Redução da probabilidade de ocorrência;

- Redução dos danos provocados por eventuais ocorrências;
- Criação de procedimentos para se recuperar de eventuais danos.

6.2.1 Política de segurança obrigatória

A política de segurança obrigatória é desenvolvida e aplicada para controlar informações com classificação ou sensibilidade, esta deve incluir regras detalhadas de como irá controlar esta informação ao longo de seu ciclo de vida. Estas regras existem em função das várias designações de sensibilidade que às informações podem assumir e, as enúmeras forma de acesso que devem ser apoiadas pelos sistemas.

Desta forma, a segurança obrigatória necessita de um conjunto de regras de controle de acesso, que restrinjam o acesso nesta informação baseado em comparação, da liberação/autorização do usuário em relação, a designação da informação, quanto a classificação/sensibilidade, controlando, com isso, o acesso.

Políticas obrigatórias tem uma implicação clara, que é de assegurar, que as designações associadas aos dados sensíveis não possam ser mudadas arbitrariamente, e de garantir que dados com sensibilidade mais alta não sejam gravados com sensibilidade mais baixa (ORANGE BOOK,1985).

6.2.2 Política de segurança discriminatória

Segurança discriminatória é o tipo principal de controle de acesso disponível em sistema de computador. A base deste tipo de segurança que é um usuário em particular, e qual a operação que será executada, sobre qual objeto específico. É possível especificar que tipo de acesso deve ser liberado, podendo a informação estar sobre domínio de outro usuário. Esta difere da segurança obrigatória, pois implementa uma política de controle baseado no que um indivíduo precisa para fazer suas tarefas.

O controle discriminatório não substitui os controles obrigatórios. Em ambientes onde a informação é classificada, a segurança discriminatória provê melhor granulação

dentro do controle de restrições globais das políticas obrigatórias. Ter acesso às informações classificadas, requer implementação de ambos tipos de controles, como condição prévia para concessão do acesso. Em geral, nenhuma pessoa pode ter acesso à informação classificada a menos que a pessoa tenha sido considerada confiável (ORANGE BOOK,1985).

6.2.3 Análise dos tipos de políticas de segurança

Conforme o (ORANGE BOOK,1985) existem dois tipos básicos de políticas que os sistemas podem se basear para controle o acesso aos dados. As políticas obrigatórias e a discriminatória, ambas apresentadas nos itens anteriores.

Estas políticas foram levantadas e comparadas com os mecanismos padrões de controle de acesso dos SGBD's, mecanismos estes apresentados nos itens e subitens de "Modelos de controle de acesso e granulação".

6.2.3.1 Aderência do controle de acesso com a política discriminatória

De acordo com (SANDHU,1993) (SANDHU,1994), o SGBD tem o controle de acesso tipo discriminatório (DAC), controlado em uma única dimensão o acesso a dados. O administrador concede aos usuários privilégios que determinam as operações (como leitura, escrita) que eles podem executar em dados. Para um usuário processar ou executar uma tarefa este deve ter privilégios apropriados para tal, como privilégio de selecionar dados de um determinado objeto, como uma tabela ou visão.

Pode ser usado, também, o mecanismo de visões do SQL o qual proporciona uma importante medida de segurança. No entanto, a abordagem que se baseia na visão é um tanto inábil, especialmente se algum usuário precisar, ao mesmo tempo, de direitos diferentes sobre subconjuntos diferentes da mesma tabela (DATE,1999).

Segundo (OSBORN,2000), os controles de acesso discriminatórios tradicionais estão provados serem inadequado para às necessidades de segurança de muitas

organizações. Da mesma forma, também são percebidos que os controles de acesso obrigatórios baseado em rótulos de segurança são impróprio para muitas situações. Nos últimos anos a noção de controle de acesso baseado em papéis (RBAC) emergiu como um candidato para preencher a lacuna entre DAC tradicional e MAC.

Mesmos que for cuidadosamente controlado a concessão do acesso a uma tabela, para um usuário com acesso SELECIONAR, este tem a possibilidade de criar uma cópia da tabela, e, assim, poder burlar todos estes controles. Além disso, mesmo que os usuários forem de confiança, é possível que isso ocorra, caso o usuário use um programa infetado por um Cavalo de Tróia que acaba fazendo tal processo (SANDHU,1994).

Em (BERTIN0,1999) é apresentado um modelo de autorização que pode ser usado para expressar várias políticas de controle de acesso discriminatório para SGBD relacional. O modelo permite autorizações positivas e negativas e apóia exceções ao mesmo tempo. O modelo é bastante flexível de acordo com o que os usuários podem especificar. Para cada autorização concedida, pode ser aplicado autorizações fortes ou permitir exceções. O mesmo prevê administração de autorização por grupos podendo usar exceções a qualquer nível da hierarquia de grupo, com isso disponibilizando suspensão temporária de autorizações. O modelo apóia a administração descentralizada de autorizações com o uso propriedade. Também, podem ser restringidos privilégios administrativos, de forma que, os donos retenham o controle sobre suas tabelas. Não há implementação deste modelo pelos SGBD's relacionais comerciais.

(ROSENTHAL,2000) desenvolveu uma proposta de extensão do processo de concessão e revogação de privilégios em SQL com o objetivo de poder impor limites aos concessores podendo reativar privilégios em períodos determinados.

De acordo com (ROSENTHAL,2000), "O modelo de segurança em SQL teve poucas extensões nos últimos 20 anos, com exceção da recente adição de controles de acesso baseado em papéis".

Diante da literatura e das pesquisas levantadas constata-se que ainda há muito que se pesquisar para conseguirmos que o processo de controle de acesso discriminatório dos SGBD's reflita às necessidades das políticas discriminatórias das empresas.

6.2.3.2 Aderência do controle de acesso com a política obrigatória

Com a pesquisa realizada constatou-se que os controles padrões dos SGBD relacionais não estão preparados para atender as políticas de segurança obrigatórias. Verificou-se que existem mecanismos que possam ser adaptados aos SGBD relacionais deixado-os com condições de proporcionar tais controles. Os fabricantes de bancos de dados estão desenvolvendo pesquisas nesta área, alguns já estão incorporando estes mecanismos que proporcionam controles de acesso com granulação mais fina.

Foram realizadas várias pesquisas com os SGBD's seguros ou multi nível que proporcionam estes controles.De acordo com (SANDHU,1993 e 1994), durante os últimos vinte anos, pesquisas consideráveis foram dedicadas no desenvolvimento de construção de bancos de dados de multi nível.

No detalhamento do sistema de controle de acesso do SGBD Oracle constatou-se que o banco de dados privado virtual (VPD), no qual cria-se uma implementação de segurança ao nível de registro própria; e o controle de acesso rótulo-baseado (label-based), no qual personaliza-se uma política de VPD, já é feito para realizar isto (LEVINGE,2002a).

Os itens abaixo são implementações ou extensões de segurança incorporada ao SGBD Oracle dando a ele suporte para as políticas de segurança obrigatórias, conforme podemos verificar abaixo.

- Visões complexas e Dinâmicas.
- Reescrever consultas de aplicação: Banco de dados Privado Virtual.
- Controle de acesso baseado em rótulo.

O controle de acesso baseado em rótulo depende da política básica de DAC, essas políticas juntas, ditam os critérios pelos quais o acesso ao objeto é permitido ou negado. Na maioria das aplicações, um número relativamente pequeno de tabelas requer um controle de acesso baseado em rótulo. A proteção provida pelo DAC basta para maioria da tabelas da aplicação (LEVINGE,2002b).

Ter acesso às informações classificadas requer implementação de ambos tipos de controles como, condição prévia para concessão do acesso. Em geral, nenhuma pessoa pode ter acesso à informação classificada a menos que a pessoa tenha sido considerada confiável (ORANGE BOOK,1985).

6.3 ANÁLISE DOS DADOS PESQUISADOS

Foram pesquisadas alternativas de tipos de controle de acesso dos SGBD's na tentativa de verificar a aderência dos mesmos às políticas de segurança das empresas. Baseado nas pesquisas realizadas montou-se o quadro abaixo para apresentar os resultados verificados.

Quadro 5 – Aderência dos controles as políticas de segurança.

Tipos de controles		**Tipos de políticas de segurança**	
		Obrigatórias	**Discriminatórias**
Padrão	Concessão de privilégio diretamente	Não adere	Adere
	Concessão de privilégio via papel	Não adere	Adere
	Uso de visões	Apenas auxilia	Adere
Oracle	Concessão de privilégio diretamente	Não adere	Adere
	Concessão de privilégio via papel	Não adere	Adere
	Uso de visões	Apenas auxilia	Adere
	VPD- Banco de dados privado virtual	Apenas auxilia	Melhora, mas necessita dos controles de papéis e diretamente.
	Controle de acesso baseado em rótulo	Adere, com auxílio dos demais	Necessita desta para garantir a obrigatória.

Pesquisas estudadas	Mecanismo de autorização flexível para SGBD relacional	Não adere	Melhora
	Extensão das operações de concessão e revogação em SQL, para limitar e reativar privilégios.	Não adere	Acrescenta controle por alguns parâmetros, podendo controlar em que horários o privilégio está ativo.
	Políticas de segurança flexíveis em SQL	Não adere	Adere
	MLS	Adere, mas necessita de outros mecanismos	Necessita desta para garantir a obrigatória

Na busca da possível validação da hipótese de centralização dos controles de acesso no SGBD e a aderência da mesma a uma política de segurança específica, apresenta-se o quadro abaixo os resultados da pesquisa na busca de exemplificar às hipóteses levantadas.

Quadro 6 - Afirmações pesquisadas.

Afirmações	**Controles de acesso dos SGBD's relacionais**		
	Padrões	**Oracle**	**Pesquisas estudadas**
Atende a política de segurança apresentada.	Não	Sim	Não
Centralização dos controles de acesso no SGBD	Sem subsídios para afirmação	Sem subsídios para afirmação	Sem subsídios para afirmação

Analisando a primeira pergunta levantada, onde especifica uma política de controle de acesso particular, e que não é das mais complexas. Com base na política e nos mecanismos de controle de acesso dos SGBD's padrões a resposta é negativa, por conseguinte a resposta da segunda pergunta também é negativa, pois a segunda dependia da primeira ser afirmativa.

Mas, comparando a primeira com as extensões apresentadas pelo SGBD Oracle há possibilidade de implementar tal política usando o mecanismo de VPD juntamente com o controle por contexto da aplicação.

De acordo com (LEVINGE,2002b), uma forma mais granular de acesso de dados é o acesso ao nível de registro. Para uma tabela qualquer com dados, ter acesso a registros específicos pode estar baseado em algumas considerações como o departamento para o qual os empregados pertencem, a responsabilidade no trabalho deles ou a sua titulação, ou outros fatores significantes.

Cada aplicação pode possuir seu próprio contexto específico de aplicação. Ao usuário não é permitido trocar arbitrariamente de contexto (LEVINGE,2002b).

Contextos de aplicações permitem um controle de acesso flexível, baseado em parâmetros, baseado em atributos de interesse da aplicação. Por exemplo, para uma aplicação de recursos humanos o contexto poderia ser a “posição”, “unidade organizacional”, e “pais” enquanto um atributo de controle de entrada de pedidos poderia ser “número do cliente” e “região de vendas” (LEVINGE,2002b).

Para segunda pergunta, os SGBD estão preparados para centralizar o controle de acesso garantindo às políticas de segurança das empresas, sem à necessidade de mecanismos adicionais? Não se encontrou subsídio suficiente para ser comprovada, apesar da primeira ser afirmativa e ser pré-requisito para comprovação da mesma.

7 CONCLUSÃO

A área dos Sistemas Gerenciadores de Base de Dados, está em contínuo aperfeiçoamento, impulsionada pela constante evolução das aplicações, tecnologias e normas. A importância que os SGBD's assumem no controle das bases de dados, conseqüentemente, o grande mercado dos mesmos, faz com que o rumo evolucionário, seja inevitavelmente influenciado por este fator. Devido, a cada vez mais, às aplicações requererem algo que não encontram satisfatoriamente em produtos derivados do modelo relacional e do sua principal linguagem associada, o SQL, os principais fornecedores de SGBD's relacionais, tentam incorporar os melhores componentes da tecnologia, procurando a cada novo lançamento adicionar extensões para fazer o controle de acesso aos dados.

"Controle de acesso" é o que tratamos neste trabalho. A função básica da proteção dos dados e do controle de acesso em um sistema é garantir a confidencialidade e a disponibilidade das informações armazenadas no SGBD.

A função de controle de acesso do sistema é utilizada para definir se o usuário tem direito a acessar ou alterar determinada informação e para garantir que apenas o usuário com esse direito pode ter acesso a essa informação.

O objetivo deste trabalho teve como tema central analisar da aderência dos controles de acessos dos sistemas gerenciadores de banco de dados com às políticas de segurança nas aplicações. Para isso, nos capítulos iniciais foram apresentados os conceitos de segurança da informação e nos seguintes foram mostrados os tipos de controles existentes, na busca da possível validação das hipóteses levantadas. Para isso optou-se em fazer o estudo dos controles básicos dos SGBDs e como especifico

utilizou-se o SGBD Oracle visto que não estava entre os objetivos o estudos das diferenças entre os controles utilizados pelos diferentes SGBDs, mas sim o a verificação da aderência destes às políticas de segurança.

Com o desenvolvimento do trabalho pode-se destacar algumas contribuições entre elas as seguintes:

Dentro das necessidades das políticas de segurança exigidas pelas empresas pode se comprovar, que os inúmeros mecanismos disponíveis nos SGBD's relacionais atuais, separadamente não conseguem serem aderentes o suficiente para atender os requisitos de segurança.

Mas, que se os administradores conhecerem melhor este controle de acesso, e utilizando-os em conjunto, pode-se ter um controle com uma granulação adequada à maioria das políticas de seguranças desenvolvidas pelas empresas.

Com todo novo trabalho surgem novas idéias para o desenvolvimento e pesquisa em novas áreas ou até mesmo no aperfeiçoamento das existentes, podemos destacas algumas dentro do contexto de controle de acesso aos dados:

- Aperfeiçoamento dos mecanismos de controle de acesso flexibilizando as concessões de privilégios, adequando-se às exigências do mercado.
- Centralização dos controles de acesso aos dados em único mecanismo diminuindo a dependência de mecanismos externos para garantia de segurança.
- Adequação dos mecanismos existentes para que estes de aderem às necessidades das políticas de segurança das empresas.
- Criação de uma interface amigável para concessão dos privilégios facilitando a administração.

8 REFERÊNCIA BIBLIOGRÁFICA

[ALBUQUERQUE,2002] ALBUQUERQUE, Ricardo. **Segurança no desenvolvimento de software**: como garantir a segurança de sistemas para seu cliente usando a ISSO/IEC. Rio de Janeiro: Ed. Campus, 2002.

[BALDWIN,1990] BALDWIN R. W. **Naming and Grouping Privileges to Simplify Security Management in Large Databases**. in Proceedings of 1990 IEEE Symposium on Research in Security and Privacy, IEEE Computer Society Press, May 1990.

[BARKER,2001] BARKER, Steve; ROSENTHAL, Arnon. **Flexible Security Policies in SQL**. IFIP Workshop on Database Security. 2001. Pages 167-180. http://citeseer.nj.nec.com/444255.html

[BERTINO,1996a] BERTINO, Elisa; JAJODIA, Sushil; SAMARATI, Pierangela. **Supporting multiple access control policies in database systems**. IEEE Transactions on Knowledge and Data Engineering, 1996.

[BERTINO,1996b] BERTINO, Elisa; JAJODIA, Sushil; SAMARATI, Pierangela; **A Non-timestamped Authorization Model for Data Management Systems**. Proceedings of the 3rd ACM conference on Computer and communications security. New Delhi, India: Pages: 169 – 178. 1996

[BERTINO,1999] BERTINO, Elisa; JAJODIA, Sushil; SAMARATI, Pierangela. **A Flexible Authorization Mechanism for Relational Data Management Systems.** ACM Transactions on Information Systems, Vol. 17, No. 2, April 1999.

[CARUSO,1999] CARUSO, Carlos A.A.; STEFFEN, Flavio Deny. **Segurança em informática e de informações.** São Paulo: Ed. Senac. 1999.

[DATE,1990] DATE, C. J. **Introdução a sistemas de bancos de dados**. Rio de Janeiro: Ed. Campus, 1990.

[DATE,2000] DATE, C. **An Introduction to Database Systems**, 7ª Editção, AddisonWesley, 2000.

[DIDRIKSEN,1997] DIDRIKSEN, Tor**. Rule Based Database Access Control - A Practical Approach.** Proceedings of the second ACM workshop on Role-based access control. Virginia, United States: Pages 143 – 151. 1997.

[FERNANDEZ,1981] FERNANDEZ, Eduardo B; SUMMERS, Rita C; WOOD, Christopher. **Database security and integrity**. Canada: Ed. Addison-Wesley Publishing Copmpany, 1981

[GIURI,1997] GIURI, Luigi; IGLIO, Pietro. **Role templates for content-based access control**. Proceedings of the second ACM workshop on Role-based access control. November 1997

[GRIFFITHS,1976] GRIFFITHS Patricia P.; WADE Bradford W. **An Authorization Mechanism for a Relational Database System**. ACM Transactions on Database Systems, Vol. 1. No. 3, September 1976, Pages 242-255.

[HAZEL,2001] HAZEL, Lorraina. **An Overview of Oracle Database Security Features.** CNE. May 13, 2001.

[KOENKE,1999] KROENKE, David M. **Bancos de dados**: Fundamentos, projeto e implementação. Rio de Janeiro: Ed. LTC Livros Técnicos e Científicos Editora S.A., 1999.

[LEVINGE,2002a] LEVINGE, Jeff. **Oracle9i Security Overview**. Part No. A96582-01. Copyright © 2001, 2002 Oracle Corporation. All rights reserved., Release 2 (9.2).

[LEVINGE,2002b] LEVINGE, Jeff. **Oracle Label Security Administrator's Guide, Release 2 (9.2).** Part No. A96578-01. Copyright © 2000, 2002 Oracle Corporation. All rights reserved.

[LUPU,1997] LUPU, Emil; SLOMAN, Morris. **A Policy Based Role Object Model**. IEEE. Published in the Proceedings of EDOC'97, October 24-26, 1997

[MOREIRA,2001] MOREIRA, Milton Stringasci. **Segurança Mínima** - Uma visão corporativa da segurança de informação. Rio de Janeiro: Ed. Axcel Books do Brasil Editora Ltda, 2001.

[ORACLE,1992] ORACLE FM. **Server** - SQL Language Reference Manual, December 1992.

[ORANGE BOOK,1985] **Orange Book**.Department of Defense Computer Security Center, Department of Defense Trusted Computer System Evaluation Criteria, Fort George G. Meade, MD 20755. December 1985.

[OSBORN,2000] OSBORN, Sylvia; SANDHU, Ravi; MUNAWER, Qamar. Configuring **Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies.** ACM Transactions on Information and System Security, Vol. 3, No. 2, May 2000, Pages 85–106.

[OZSU,1999] OZSU, M. Tamer; VALDURIEZ, Patrick. **Principles of distributed database system**. New Jersey: Ed. Prentice Hall,1999.

[ROSENTHAL,2000a] ROSENTHAL, Arnon; SCIORE, Edward. **How Can Data Sources Specify Their Security Needs to a Data Warehouse**. http://citeseer.nj.nec.com/450453.html. 2000

[ROSENTHAL,2000b] ROSENTHAL, Arnon; SCIORE, Edward; **Extending SQL's Grant and Revoke Operations, to Limit and Reactivate Privileges**. IFIP Workshop on Database Security. 2000

[SANDHU,1993] SANDHU, Ravi S. and JAJODIA, Sushil. **Data And Database Security And Controls**. Handbook of Information Security Management, Auerbach Publishers, 1993, pages 481-499.

[SANDHU,1994] SANDHU, Ravi S. **Relational Database Access Controls**. Handbook of Information Security Management (1994-95 Yearbook), Auerbach Publishers, 1994, pages 145-160.

[SANDHU,1996] SANDHU, Ravi S; COYNE, Edward J; FEINSTEINK, Hal L. **Role-Based Access Control Models**. IEEE Computer, Volume 29, Number 2, February 1996, pages 38-47. Revised October 26, 1995

[SANDHU,1997] SANDHUY, Ravi S. **Role-Based Access Control** Laboratory for Information Security Technology. ISSE Department, MS 4A4, September 17, 1997

[SILBERSHATZ,1999] SILBERSCHATZ, Abraham; KORTH, Henry F; SUDARSHAN, S. **Sistema de banco de dados**. São Paulo: Ed. Makron Books, 1999.

[SIMON,1997] SIMON, Richard; ZURKO, Mary Ellen. **Separation of duty in role-bvased environments**. IEEE Computer Security Foundations Workshop.1997.

[SMITH,1992a] SMITH, Kenneth; WINSLETT, Marianne. **The importance of declarative semantics for MLS ralational database.** citeseer.nj.nec.com/102742.html. 1992

[SMITH,1992b] SMITH, Kenneth; WINSLETT, Marianne. **Multilevel secure rules:** Integrating the multilevel secure and active data models. IFIP Workshop on Database Security. 1992

[SQUADRA,2002] **Sistema de segurança** http://www.squadra.com.br/produtos/guardiantec.htm em 11/05/02

[TIDSWELL ,1999] TIDSWELL, Jonathon E.; OUTHRED, Geoffrey H.; POTTER, John M. **Dynamic Rights: Safe Extensible Access Control** ACM Workshop on Role-Based Access Control. **1999**

[WINSLETT,1994] WINSLETT, Marlanne. **Formal Query Languages for Secure Relational Databases**. ACM TransactIons on Database Systems, Vol. 19, No. 4, Pages 626-662. December 1994.

[DENNING,1979] DENNING, Dorothy E.; DENNING Peter J. **Data Security***. ACM Computing Surveys, Vpl. II, No. 3, September 1979

[PISSINOU,1994] PISSINOU, Niki; MAKKI, Kia; PARK, E.K.. **Towards a Framework for Integrating Multilevel Secure Models and Temporal Data Models**. ACM Computing Surveys. 1994

[GLOBONEWS,2001] GLOBONEWS. **Batalha dos conversores entre IBM, Oracle e Microsoft.** http://globonews.globo.com/GloboNews/article/0%2C6993%2CA86011-19%2C00.html. Quinta-feira, 09/08/2001 - 12h14m. Acessado em 12/10/2003.

[PROCESSOR,2001] PROCESSOR. **Oracle lança nova versão de Banco de Dados**. http://www.processor.com.br/news/abrenewshist.asp?News=562. quarta-feira, 01/08/200. Acessado em 12/10/2003.